НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Інститут телекомунікаційних систем

Кафедра інформаційно-телекомунікаційних мереж

«На правах рукопису»
УДК 004.021

«До захисту допущено»
Завідувач кафедри
_____ Лариса ГЛОБА
«___»_____2020 р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою «Інформаційно-комунікаційні технології»
зі спеціальності 172 «Телекомунікації та радіотехніка»
на тему: «Удосконалений спосіб функціонування гіпервізора NFV в мережах SDN»

Виконав:
студент ІІ курсу, групи ТІ-91мп
Пархоменко Дмитро Олександрович _____

Керівник:
Професор кафедри ІТМ ІТС, д.т.н.
Скулиш Марія Анатоліївна _____

Рецензент:
Доцент кафедри ТК ІТС, к.т.н., с.н.с.
Міночкін Дмитро Анатолійович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2020

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут телекомунікаційних систем
Кафедра Інформаційно-телекомунікаційних мереж

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інформаційно-комунікаційні технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Лариса ГЛОБА

«___»_____2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Пархоменку Дмитру Олександровичу

1. Тема дисертації «Удосконалений спосіб функціонування гіпервізора NFV в мережах SDN», науковий керівник дисертації професор кафедри інформаційно-телекомунікаційних мереж ІТС Скулиш Марія Анатоліївна, д.т.н., затверджені наказом по університету від «03» листопада 2020 р. № 3208-с

2. Термін подання студентом дисертації 10.12.2020 р.

3. Об’єкт дослідження: Спосіб функціонування гіпервізора NFV в мережах SDN

4. Предмет дослідження: Методи віртуалізації мережевих функцій в програмно керованих мережах

5. Перелік завдань, які потрібно розробити:

1. Провести аналіз існуючих методів віртуалізації у традиційних мережах.

2. Провести аналіз існуючих методів віртуалізації у програмно-конфігурованих мережах.

3. Для методів Vertigo, OVX, FlowVisor ,використовуючи програмне забезпечення MiniNet, створити імітаційну модель з метою проведення експериментального дослідження для їх оцінки та порівняння.

4. Розгортання фізичної топології для тестування часу налаштування потоку та пропускної здатності між хостами.

5. Аналіз отриманих результатів. Розробка рекомендацій щодо використання удосконаленого способу функціонування гіпервізора NFV.

6. Орієнтовний перелік ілюстративного матеріалу:

1. Тема, мета, актуальність, задачі дослідження.

2. Аналіз існуючих підходів щодо методів віртуалізації у традиційних та програмно-конфігурованих мережах .

3. Удосконалений спосіб функціонування гіпервізора NFV в мережах SDN.

4. Результати експерименту та імітаційного моделювання роботи методів віртуалізації у ПКМ.

5. Загальні висновки.

7. Орієнтовний перелік публікацій

8. Дата видачі завдання 01.09.2019 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Вибір та обґрунтування напрямку дослідження	01.09.2019 – 30.09.2019	виконано
2	Дослідження використання технології SDN та NFV, їх взаємозв'язок	01.10.2019 – 30.11.2019	виконано
3	Порівняльна оцінка існуючих методів віртуалізації у традиційних мережах	01.12.2019 – 28.02.2020	виконано
4	Оцінка аналіз існуючих методів віртуалізації у програмно-конфігурованих мережах	01.03.2020 – 15.04.2020	виконано

5	Опис та постановка задачі для розгортання топології віртуальних мереж	16.04.2020 – 31.05.2020	виконано
6	Апробація теоретичних результатів дослідження	1.06.2020 – 30.06.2020	виконано
7	Підготовка середовища, планування та проведення експерименту з метою отримання вихідних даних	1.07.2020 - 30.09.2020	виконано
8	Аналіз отриманих результатів та формування практичних рекомендацій	01.10.2020 - 30.11.2020	виконано
9	Підготовка звітної документації	01.12.2020 - 21.12.2020	виконано

Студент Дмитро ПАРХОМЕНКО

Науковий керівник дисертації Марія СКУЛИШ

РЕФЕРАТ

Робота містить 105 сторінок, 35 рисунків та 2 таблиці. Було використано 34 джерело.

Актуальність: на сьогоднішній день активно розвиваються мережеві технології з використання розумних систем керування. А саме за останні 5 років все більше компаній пропонують пакетні рішення для організації транспорту у телекомунікаційній мережі. Це стає можливим за рахунок використання програмно керованих мереж(SDN) та технології віртуалізації мережевих функцій. Однак багато областей залишаються невивченими сучасними реалізаціями гіпервізора мережі. У цій дипломній роботі представлено та оцінено деякі функції, що надаються мережевими гіпервізорами, такі як повна доступність простору заголовків, ізоляція та прозорі можливості пересилання трафіку для орендарів та інше. Результати вказують на те, що гіпервізори мережі привносять гнучкість SDN у процес віртуалізації мережі, що полегшує мережевим адміністраторам розподіл мережі між орендарями. Однак така підвищена гнучкість може спричинити за собою зниження продуктивності, а також створює додаткові ризики взаємодії через відсутність стандартизації методів віртуалізації.

Мета роботи: розробити рекомендації щодо функціонування гіпервізору NFV за рахунок використання ряду засобів реалізації контролера SDN що дозволить виявити переваги та недоліки існуючих рішень та сформувати пакетне рішення для конкретної задачі побудови мережі.

Задачі дослідження:

1. Провести огляд існуючих методів віртуалізації у традиційних мережах.
2. Провести аналіз існуючих методів віртуалізації у програмно-конфігурованих мережах.

3. Для методів Vertigo, OVX, FlowVisor ,використовуючи програмне забезпечення MiniNet, створити імітаційну модель з метою проведення експериментального дослідження для їх оцінки.
4. Розгортання фізичної топології для тестування часу налаштування потоку та пропускної здатності між хостами.
5. Аналіз отриманих результатів. Розробка рекомендації щодо функціонування гіпервізору NFV за рахунок використання ряду засобів реалізації контролера SDN що дозволить виявити переваги та недоліки існуючих рішень та сформувати пакетне рішення для конкретної задачі побудови мережі.

Об'єкт дослідження: процес віртуалізації мережевих функцій в програмно-конфігурованих мережах.

Предмет дослідження: функціонування гіпервізору NFV за рахунок використання ряду засобів реалізації контролера SDN.

Методи дослідження: основними методами дослідження є математичне та імітаційне моделювання.

Наукова новизна: Запропонували удосконалений спосіб функціонування гіпервізора NFV в мережах SDN, який базується на групі методів віртуалізації та дозволяє використати переваги кожного з них.

Практична новизна: розроблено ряд коротких програм (скриптів) які дозволяють проводити тестування імітаційних моделей ПКМ з віртуалізацією.

Ключові слова: SDN, NFV, програмно-конфігурована мережа, віртуалізація мережевих функцій, FlowVisor, VeRTIGO, OpenVirteX

ABSTRACT

The work contains 105 pages, 35 figures and 2 tables. 34 sources were used.

Relevance: today, network technologies for the use of smart control systems are being actively developed. Namely, over the past 5 years, more and more companies offer package solutions for the organization of transport in the telecommunications network. This is made possible by the use of software-controlled networks (SDN) and virtualization technology for network functions. However, many areas remain unexplored by modern network hypervisor implementations. This thesis presents and evaluates some of the features provided by network hypervisors, such as full availability of header space, isolation and transparent ability to forward traffic to tenants, and more. The results indicate that network hypervisors add SDN flexibility to the network virtualization process, making it easier for network administrators to distribute the network between tenants. However, such increased flexibility can lead to reduced productivity, as well as create additional risks of interaction due to the lack of standardization of virtualization methods.

Purpose: to develop recommendations for the operation of the NFV hypervisor through the use of a number of tools for the implementation of the SDN controller that will identify the advantages and disadvantages of existing solutions and form a package solution for a specific task of building a network.

Research objectives:

1. Review existing methods of virtualization in traditional networks.

2. To analyze the existing methods of virtualization in software-configured networks.

3. For Vertigo, OVX, FlowVisor methods, using MiniNet software, create a simulation model to conduct an experimental study to evaluate them.

4. Deploy a physical topology to test flow tuning time and bandwidth between hosts.

5. Analysis of the results. Development of recommendations for the operation of the NFV hypervisor through the use of a number of tools to implement the SDN controller that will identify the advantages and disadvantages of existing solutions and form a package solution for a specific task of building a network.

Object of research: the process of virtualization of network functions in software-configured networks.

Subject of research: functioning of the NFV hypervisor due to the use of a number of means of implementation of the SDN controller.

Research methods: the main research methods are mathematical and simulation.

Scientific novelty: An improved way of NFV hypervisor operation in SDN networks has been proposed, which is based on a group of virtualization methods and allows to take advantage of each of them.

Practical novelty: a number of short programs (scripts) have been developed that allow testing of PKM simulation models with virtualization.

Keywords: SDN, NFV, software-configured network, virtualization of network functions, FlowVisor, VeRTIGO, OpenVirteX

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

SDN	software-defined networking
NFV	Network Functions Virtualization
ПКМ	програмно-керована мережа
OVX	OpenVirteX
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
IANA	Internet Assigned Numbers Authority
OVS	Open vSwitch
FL	Floodlight
FV	FlowVisor

ВСТУП

У міру зростання тенденції до програмно-визначених мереж в останні роки було розроблено кілька контролерів віртуалізації мережі. Ці контролери, також звані мережевими гіпервізорами, намагаються керувати фізичними мережами на основі SDN, щоб кілька орендарів могли безпечно, спільно використовувати одне і те ж обладнання в площині пересилання без ризику впливу інших орендарів. Однак багато областей залишаються недослідженими в поточних реалізаціях мережевих гіпервізора. У цій дисертації представлені і оцінені деякі функції, пропоновані мережевими гіпервізорами, такі як доступність повного простору заголовків, ізоляція і прозора пересилання трафіку для клієнтів. Час налаштування потоку і пропускна здатність також вимірюються і порівнюються між різними мережевими гіпервізорами.

Оцінюються три різних мережевих гипервизора: FlowVisor, VeRTIGO і OpenVirteX. Ці інструменти віртуалізації оцінюються за допомогою експериментів, що проводяться на трьох різних випробувальних стендах: емульований сценарій Mininet, фізичний випробувальний стенд з одним комутатором, а також віддалений випробувальний стенд GENI. Результати показують, що мережеві Гіпервізор привносять гнучкість SDN в віртуалізацію мережі, спрощуючи для мережевих адміністраторів точне визначення того, як мережа ділиться на частини і ділиться між клієнтами. Однак така підвищена гнучкість може призвести до зниження продуктивності, а також до додаткових ризиків взаємодії через відсутність стандартизації методів віртуалізації.

РОЗДІЛ 1
SDN ОСНОВНІ ПОЛОЖЕННЯ, КОНЦЕПЦІЯ

1.1 Введення і визначення понять.

Традиційні телекомунікаційні мережі проектувалися з розрахунку на використання спеціалізованих апаратних пристроїв (маршрутизаторів, Ethernet-комутаторів, обладнання EPC (Enhanced Packet Core), міжмережевих екранів, балансувальників навантаження тощо. Ці пристрої створювалися на базі специфічних апаратних і програмних платформ окремих вендорів. Розгортання цих «монолітних» мережевих елементів призводило до тривалих циклів проектування і запуско-налагоджувальних робіт, а, отже до уповільнення виведення на ринок нових продуктів і послуг. Обслуговування та управління такою мережею було досить неефективним і дорогим. Все це призводило до того, що зростання інвестицій в розвиток мережі для задоволення запитів абонентів перевищувало зростання доходів від надання послуг в ній.

Тим не менш, у даний час мережі телекомунікаційних операторів складаються у основному з «монолітних» мережевих елементів, де функції управління, адміністрування і пересилання даних (трафік даних користувача) виконуються на основі фізичних, "залізних" пристроїв. Дуже часто мережу будується з мережевих елементів від одного виробника (вендора), оскільки в цьому випадку, дійсно легше забезпечити сумісність. Політика "ексклюзивного постачальника" є загальноприйнятою в вендорському середовищі. Розгортання нових послуг, модифікації ("апгрейди") обладнання або послуг робиться по черзі на кожному мережевому елементі і вимагає тісної координації внутрішніх і зовнішніх ресурсів оператора. Така монолітна організація робить операторську мережу негнучкою, ускладнює введення нових послуг і функцій, а також збільшує залежність оператора від специфічних ("пропрієтарних") рішень конкретних вендорів.

Тому, в даний час багато операторів вибрали шлях цифрової трансформації на базі технологій SDN / NFV. Задачі такої трансформації - такі:

1. Підвищення операційної ефективності:

- Досягнення еластичності і масштабованості в масштабі всієї мережі оператора.
- Автоматизація операцій (Operation), адміністрування (Management) і обслуговування (Maintenance), OAM.
- Динамічне управління потоками трафіку, з відповідним перерозподілом мережевих ресурсів в "реальному часі".
- Оперативне створення послуг з ланцюжків сервісів.

2. Трансформація бізнес-моделі:

- Зниження часу виведення послуг на ринок .
- Усунення локальних рішень, досягнення можливості швидкого впровадження інновацій в масштабі всієї мережі.
- Швидке і ефективне створення і надання послуг (Agile).

Основні властивості мережі, побудованої на принципах SDN / NFV:

- Поділ площини управління і передачі даних.
- Віртуалізація мережевих функцій.
- Програмоване управління мережевими ресурсами, обчислювальними ресурсами, і ресурсами зберігання даних, а також оркестрації послуг.
- Стандартизація протоколів і автоматизація конфігурації мережевих елементів.
- Єдиний механізм адміністрування і виділення ресурсів мережі за запитом для різних послуг і функцій.

Комплексне використання цих нових властивостей дозволяють реалізувати динамічне підлаштування мережі під потреби додатків, що підвищує операційну гнучкість і спрощує розгортання послуг.

1.2 Програмно–конфігуровані мережі. Основні положення.

Традиційні мережі - це розподілені системи, що складаються з мережевих пристроїв, які відповідають за обробку як площини управління, так і площини пересилання даних мережі. Функція площини управління вирішує, як слід обробляти трафік мережі; вона відповідає за виявлення сусідніх пристроїв та мереж та вирішення питання, куди слід перенаправляти трафік. Протоколи виявлення топології, дерева охоплення та маршрутизації є прикладами функцій площини управління [33]. Площина пересилання даних використовує лише таблиці пересилання, які раніше були розраховані площиною управління, і використовує цю інформацію для швидкого пересилання трафіку [33].

Програмно–конфігурована мережа (SDN, Software-defined Networking) - мережа передачі даних, в якій рівень управління мережею відділений від пристроїв передачі даних і реалізується програмно [1].

SDN – це нова концепція, яка має на меті зняти функції площини управління з мережевих пристроїв і розмістити їх у спеціалізованому, логічно централізованому контролері мережі. Це дає перевагу в спрощенні конфігурації мережі та дотримання правил. Це також зменшує необхідні системні ресурси, оскільки вони більше не потрібні для запуску функцій площини управління [33].

Концепція SDN передбачає:

– відокремити в маршрутизаторі управління мережевим обладнанням від управління передачею даних. Управління винести на окремий комп'ютер, який буде знаходитися під контролем адміністратора мережі;

– перейти від управління окремими екземпляром мережевого обладнання до управління мережею в цілому;
– створити інтелектуальний програмно-керований інтерфейс між мережним додатком і транспортним середовищем (Рис. 1.1) [2].

Таким чином, реалізація концепції SDN – розділення управління мережею (площини управління) і механізму просування даних (площини даних), перенесення функцій управління в окремі обчислювальні пристрої, які називаються SDN-контролерами, приводить до заміни традиційної розподіленої моделі маршрутизації централізованою моделлю, перетворюючи процес управління мережею, що включає створення маршрутів, в процес програмування мережі в цілому [3].

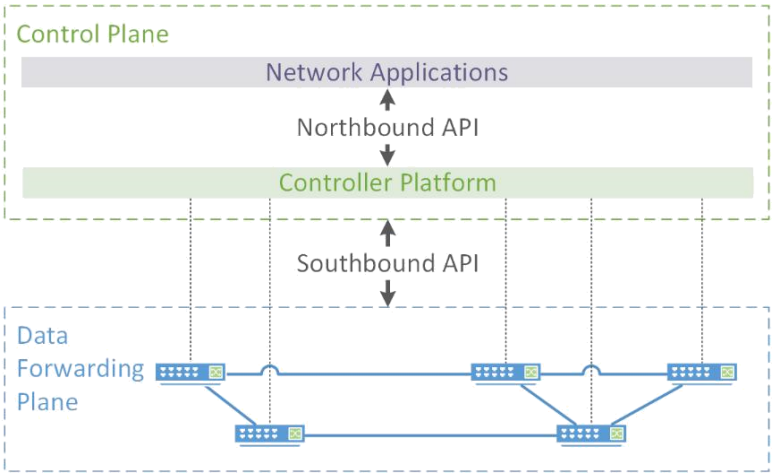


Рис. 1.1 Спрощена архітектура мережі SDN

В архітектурі SDN точкою відліку, як правило, є централізована платформа контролера, зображена в центрі Рисунка 1.1. Мережеві додатки - це програмні модулі, які взаємодіють із платформою контролера через інтерфейс програмування додатків (API), програмний інтерфейс. Типовими мережевими додатками є «MAC learning», балансування навантаження та алгоритми маршрутизації. API програмного інтерфейсу не дуже добре стандартизовані, і є

безліч різних варіантів на вибір [33]. Тому платформа контролера та мережеві програми часто інтегровані в єдине програмне забезпечення, яке називається контролером. POX, Floodlight та Beacon - це приклади контролерів, які пропонують стандартну програму «MAC learning».

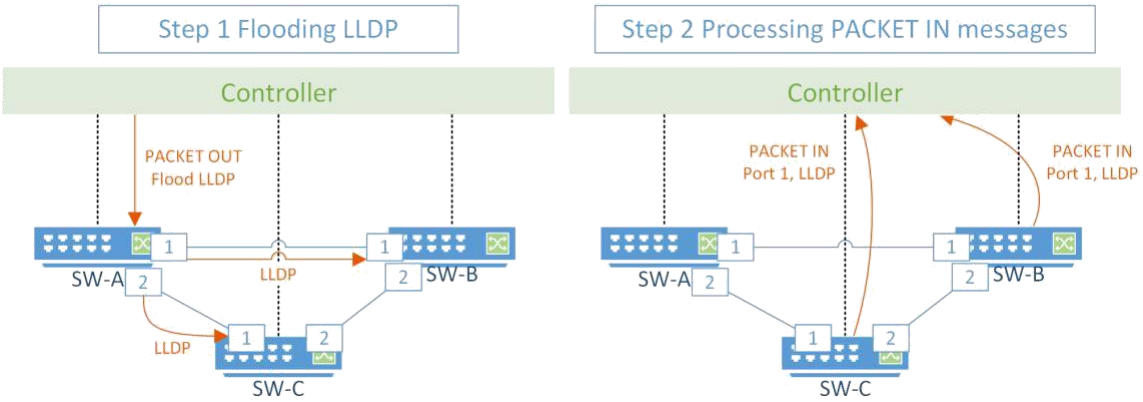
Площина пересилання даних складається з простих пристроїв переадресації, які зазвичай називаються комутаторами, які зв’язуються з контролером через API. Протокол OpenFlow є найбільш відомим, стандартизованим та активно використовуваним програмним інтерфейсом API для SDN.

OpenFlow дозволяє контролеру маніпулювати трафіком у площині пересилання даних шляхом встановлення правил потоку в таблицях потоків OpenFlow комутаторів. Кожне правило потоку складається з пар, збігів та дій. Наприклад, пакет може бути зіставлений за MAC-адресою призначення, IP-адресою джерела та / або багатьма іншими типовими полями заголовка. Якщо пакети відповідають правилу потоку, тоді їх можна переадресувати контролеру, пропустит, перезаписати заголовки, відправити з усіх портів комутаторів тощо [33].

Ключовою функцією віртуалізації мережі в SDN є дослідження топології. Для успішної реалізації переадресації трафіку контролеру необхідно знати топологію площини переадресації даних - як фізично взаємопов’язані мережеві пристрої. Зазвичай контролери OpenFlow виявляють площину пересилання даних, спочатку надсилаючи повідомлення OpenFlow PACKET OUT з проханням мережевих пристроїв залити кадри протоколу виявлення рівня зв'язку (LLDP) з усіх портів. Потім контролер чекає повідомлень OpenFlow PACKET IN, що несуть повідомлення LLDP. Повідомлення PACKET IN містять інформацію про те, який порт отримав які повідомлення LLDP, що дозволяє контролеру будувати графік мережі. Цей процес пояснюється Pakzad et al. [47] і

зображено на малюнку 1.2. Контролер повністю виявляє топологію, повторюючи цей процес для всіх мережевих пристроїв

Рис. 1.2 Виявлення топології за допомогою затоплення LLDP



Ідентифікація пристрою також відіграє важливу роль у віртуалізації мережі на основі SDN. Кожен комутатор OpenFlow, що підтримує мережу, однозначно ідентифікується за допомогою ідентифікатора шляху передачі даних (DPID), 64-бітового поля, що складається з 48 бітів реальної унікальної MAC-адреси мережевого пристрою плюс 16 бітів, які залишаються як додаткове поле ідентифікації. Постачальники можуть використовувати це додаткове 16-бітове поле будь-яким способом, яким вони бажають [8]. Наприклад, у комутаторі Hewlett-Packard, кожен екземпляр OpenFlow асоціюється з номером VLAN у комутаторі. Додаткове 16-бітове поле в DPID використовується для перенесення номера VLAN, який відповідає екземпляру OpenFlow. Це дозволяє легко розрізнити, якими портами комутатора можна керувати через певний DPID - будь-які порти, що належать до даного VLAN.

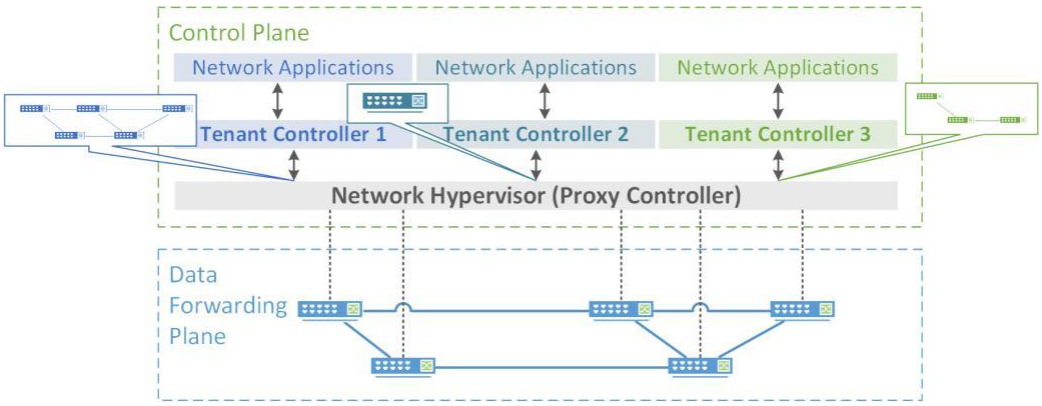


Рис. 1.3 Мережевий гіпервізор в архітектурі SDN

Віртуалізація на основі SDN зазвичай реалізується шляхом введення контролерів проксі-серверів, які діють як гіпервізори мережі між площиною пересилання даних та багатьма контролерами-орендарями, які спільно використовують контроль над однією і тією ж інфраструктурою. Як показано на малюнку 1.3, гіпервізори мережі можуть змінити спосіб бачення мережі контролерами орендаря. Контролер орендаря 1 має прозорий контроль над усіма мережевими пристроями; Контролер оренди 2 бачить абстракцію всієї мережі, представлену одним великим комутатором; Контролер оренди 3 бачить лише частину мережі, що містить три з п’яти комутаторів.

Прикладами мережевих гіпервізорів, які працюють за принципом проксі-контролера, є: FlowVisor, OpenVirteX, VeRTIGO , AutoVFlow та AutoSlice.

Важливо спочатку зрозуміти хоча б деякі найосновніші традиційні технології віртуалізації перед подальшим вивченням мережевої віртуалізації на основі SDN. Традиційні методи, деякі з яких будуть коротко висвітлені в наступних кількох розділах, можуть бути використані для подальшого вдосконалення нових методів, реалізованих за допомогою SDN.

1.3 Обмеження традиційних мережей

Традиційні методи розвитку мереж полягають в «апаратному» принципі: «пристрій - функція». У більшості випадків для того, щоб ввести чергову функцію, треба встановити інший пристрій. Однак, такі методи розвитку мереж і виведення на ринок нових послуг за допомогою нових фізичних пристроїв стикаються з рядом обмежень.

Обмеження гнучкості

Пропріетарний вендорських дизайн обладнання полягає в тому, що кожний мережевий пристрій має фіксовану комбінацію обладнання та програмного забезпечення з невеликими варіаціями. Це дуже обмежує операторів у виборі функціональних комбінацій і можливостей обладнання, які можуть бути розгорнуті на мережі.

Обмеження масштабованості

Можливості масштабування фізичних мережевих пристроїв обмежені як в частині обладнання, так і програмного забезпечення. Устаткування вимагає живлення і місця для розміщення, що часто обмежене в зонах щільної забудови.

Довгий час виведення послуг на ринок

Вимоги ринку ростуть, але зростання кількості функцій і одиниць обладнання не завжди здатні встигати за цими вимогами. Часто введення нових функцій вимагає апгрейда існуючого і введення нового обладнання .. Все це значно збільшує капітальні витрати і вартість володіння інфраструктурою мережі.

Обмеження адміністрування

Системи моніторингу використовують стандартні протоколи, такі як SNMP, NetFlow, Syslog і т.п. для збору інформації про стан пристроїв. Однак для моніторингу вендороспеціфічних (пропрієтарних) параметрів, стандартних систем може бути недостатньо. В цьому випадку, для кожного мережевого

домену, побудованого на обладнанні певного постачальника, потрібна пропріетарна система моніторингу.

Операційні витрати

Вендороспеціфічне обладнання підвищує операційні витрати, оскільки вимагає наявності в штаті оператора відповідних фахівців. Або, як варіант, вимагає використання «Managed services» (тобто «професійних послуг» від постачальника). Це призводить до «вендорозалежності», тобто, до залежності до рішень конкретного постачальника.

Плавність зростання ємності мережі

Вимоги до зростання ємності мережі (як на короткі, так і на тривалі терміни) складно спрогнозувати. Це веде до непродуктивних капітальних витрат. Навпаки, коли ресурси мережі виявляються вичерпаними, проходить чимало часу, перш ніж ємність мережі вдається розширити.

Взаємодія

Часто буває так, що для прискорення виведення обладнання на ринок постачальники оснащують його новими функціями до повного завершення процесу їх стандартизації. У багатьох випадках це призводить до несумісності різнорідного обладнання на мережі, необхідність випробувань мережевих рішень в лабораторіях операторів і сервіс-провайдерів перед тим, як розгортати ці пристрої на мережі.

Віртуалізація - технологія яка дозволяє запускати кілька операційних систем на одному фізичному сервері. Концепція віртуалізації серверів досить давно використовується в дата-центрах (ЦОД, центрах обробки даних). При цьому, фізичні сервери замінюються їх віртуальними «копіями», що працюють поверх гіпервізора. Це дозволяє, крім іншого, досягти більш ефективного використання фізичних ресурсів дата-центру.

NFV можна визначити як метод і технологію, яка дає можливість замінити фізичні мережеві пристрої з певними функціями на програмні суті, виконують такі ж функції на загальнодоступному серверному обладнанні.NFV розширює концепцію віртуалізації, крім серверів, також і на всі типи мережевих пристроїв.

Фактично, NFV відокремлює програмне забезпечення від обладнання, і надає можливість використовувати будь-який комерційно доступне, стандартне обладнання COTS (Commercial Off the Shelf) для виконання на ньому спеціалізованих мережевих функцій, які можна змінювати швидко і в будь-який момент.

1.4 Переваги NFV

На початку були коротко описані обмеження традиційних методів розвитку мереж зв'язку. Розглянемо, як віртуалізація мережевих функцій NFV вирішує більшість цих обмежень, а також привносить додаткові переваги. Багато з того, що в традиційній мережі оператора було нереалізованим, і тому такі можливості навіть не розглядалися, стає можливим в NFV.

Свобода вибору обладнання

Оскільки NFV використовує звичайне, комерційно доступне комп'ютерне обладнання COTS, оператори можуть вибирати найбільш підходяще за цінами і підтримки обладнання від численних виробників, і таким чином, найбільш оптимально будувати свої мережі, як за витратами, так і функціоналом.

Можна сказати, що все розмаїття традиційного мережевого обладнання, що поставляється вендорами, в NFV зводиться лише до трьох його видів: сервер, система зберігання, мережеві пристрої. Однак, число постачальників такого стандартного устаткування набагато більше, ніж спеціалізованого телекомівські. Весь функціонал при цьому забезпечується програмними функціями, які працюють на цій обмеженій номенклатурі устаткування. Ці

функції зазвичай розробляються вендорами незалежного ПО, або самими операторами.

Процес модифікації традиційного обладнання на мережі оператора зв'язку зазвичай буває дуже довгим і витратним. З NFV, оператори можуть вводити нові функції за кілька годин або навіть хвилин з панелі управління адміністратора, а не розгортати на мережі нові пристрої з залученням висококваліфікованого технічного персоналу.

Наприклад, при необхідності розширення ємності Інтернет-шлюзу, замість установки нових плат в кошик блейд-сервера, конфігурації, модифікації таблиць та ін., Оператор просто може призначити нові віртуальні машини з наявного пулу ресурсів, на яких будуть запущені відповідні VNF.

Швидкість і оперативність

На противагу фізичного обладнання, мережеві функції VNF можуть створюватися і видалятися «на льоту», здебільшого автоматизовано, без необхідності залучення роботи технічного персоналу.

Така властивість носить назву «еджайл» (agile - гнучкий, оперативний, моторний, ефективний), термін, який укорінився в технічній літературі в англійській транскрипції, оскільки його неможливо перевести на російську мову одним словом.

Масштабованість і еластичність

Введення нових послуг і додатків, які вимагають значної смуги пропускання мережі, в сьогоднішніх умовах часто змушують операторів працювати в постійному стресі, щоб задовольнити все зростаючі потреби абонентів і користувачів в нових послугах так, щоб вони нормально працювали на існуючих ресурсах. Традиційні ресурси часто являють собою «пляшкове горлечко» (bottleneck), коли всі інші ресурси мережі дозволяють надати новий

сервіс без проблем, але фізичні мережеві елементи одного-двох ресурсів є недостатніми, а розширювати їх - довго і накладно.

Ця проблема вирішується в NFV, яка дозволяє отримувати потрібні ресурси дуже швидко, розгортаючи нові VNF на віртуальних машинах VM з наявного пулу ресурсів.

Оскільки ці VNF не обмежені параметрами спеціалізованого фізичного обладнання, вони можуть забезпечити властивість «еластичності», тобто, вони можуть бути розгорнуті, коли вони потрібні, і згорнуті, коли вони не потрібні.

Крім того, це дозволяє уникнути звичайної ситуації в традиційній мережі, коли одні мережеві елементи перевантажені, а інші недовантажені. Швидке розгортання VNF дозволяє рівномірно розподілити наявну в даний момент навантаження.

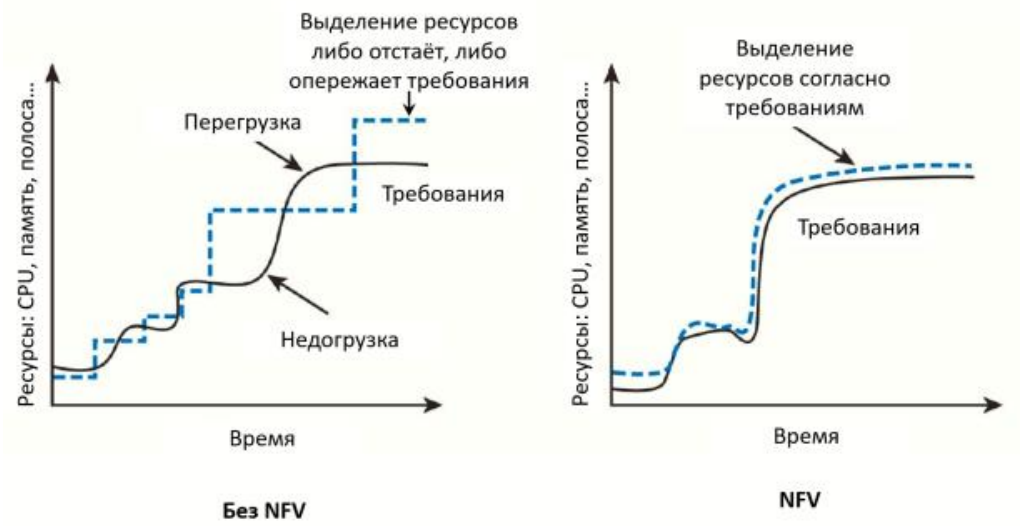


Рис.1.4 Еластичність NFV.

Використання стандартних IT-засобів

Оскільки NFV використовує ту ж саму інфраструктуру, що і стандартні дата-центри, вона може використовувати всі напрацьовані в них прийоми

розгортання і управління. Це дає можливість використовувати існуючі методи і засоби IT для телекомунікаційних мереж.

Швидке розгортання і позбавлення від вендорозалежності

Внаслідок того, що NFV забезпечує засоби швидкого розгортання стандартних рішень без надмірних витрат, пов'язаних з «моновендорськими» рішеннями, оператори можуть позбутися т.зв. «Вендорозалежності», тобто, надмірну прив'язаність до спеціалізованих рішень невеликої кількості вендорів на мережі.

Нові рішення можуть бути розгорнуті на мережі швидко, без необхідності очікування розробки нових функцій від традиційних вендорів, що часто займає тривалий час.

Крім того, це дозволяє швидко розгортати нові рішення на випробувальних доменах, протестувати їх, і в разі успішності випробувань, також швидко розгортати їх на живій мережі. У разі невдачі, їх вартість мінімальна, оскільки такі випробування пов'язані тільки з установкою і запуском програмного забезпечення, а не з придбанням і запуском нового дорогого обладнання.

Спрощення обслуговування та технічної експлуатації

Обслуговування та підтримка операцій на за допомогою NFV дозволяє знизити можливі періоди недоступності послуг. Наприклад, вихід з ладу віртуальної машини, на якій працює функція мережі негайно спричинить за собою запуск резервної віртуальної машини, яка буде виконувати VNF точно з того ж місця, на якому стався збій активної VM.

Це дозволяє також досягати модифікації програмного забезпечення в процесі роботи, ISSU (In-Service Software Upgrade) в режимі 24/7.

Все це значно знижує і навіть повністю усуває втрати, пов'язані з несправностями на мережі.

1.5 Структура (Framework) NFV

Термін NFV вперше був введений провідними операторами зв'язку світу на SDN OpenFlow World Congress в 2012 році. Вони проаналізували обмеження традиційного методу розвитку мережі, зазначені вище, і створили робочу групу по розробці специфікацій NFV ISG (Industry Specification Group) під керівництвом Європейського інституту по розробці стандартів для телекомунікацій ETSI (European Telecommunications Standards Institute).

Робоча група NFV ISG висунула три основних критерії, які повинні бути реалізовані в стандартах (рекомендаціях) для NFV:

- Розділення (Decoupling): повне розділення обладнання та програмного забезпечення.
- Гнучкість (Flexibility): автоматизоване і масштабоване розгортання мережевих функцій.

Динамічні операції (Dynamic operations): контроль за операційними параметрами мережі за допомогою точного (гранулярного) управління і моніторингу стану мережі.

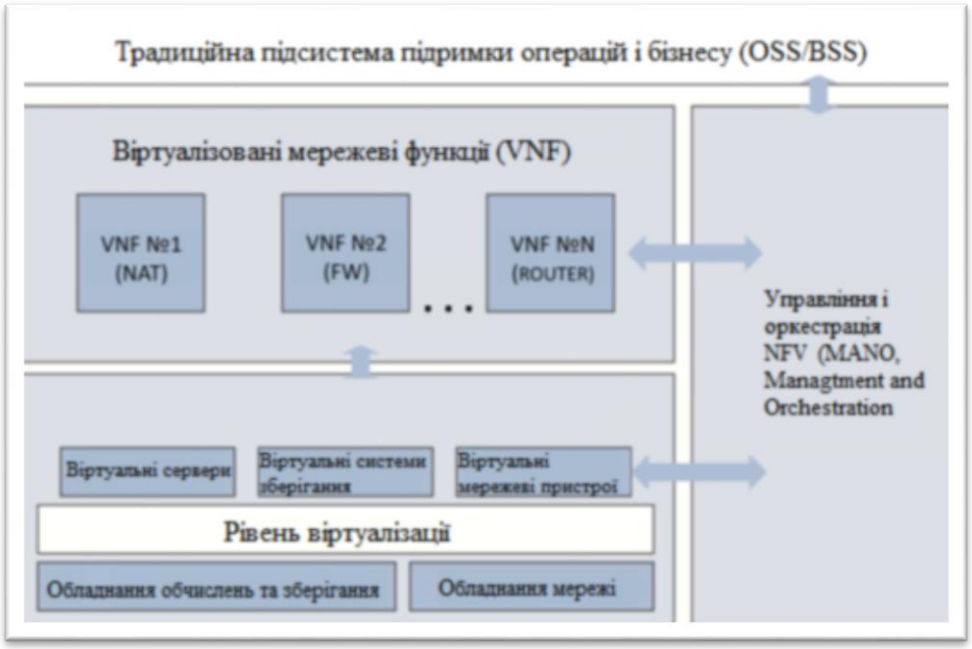


Рис. 1.5 Узагальнена архітектура NFV

Архітектура NFV складається з трьох основних підсистем:

- Віртуалізувані мережеві функції VNF (Virtualized Network Function)
- Інфраструктура віртуалізації NFVI (NFV Infrastructure)
- Підсистема управління та оркестрації MA- (Management and Orchestration)

Крім NFVI, підсистеми являють собою програмне забезпечення, а не обладнання. NFVI включає в себе як фізичне устаткування (обчислень, зберігання, мережі), так і віртуальне «обладнання»: сервери, системи зберігання, мережеві пристрої. Рівень віртуалізації (Гіпервізор і гостьові операційні системи) дають можливість розгортати на фізичних серверах віртуальні машини VM (Virtual Machines), які виконують будь-які покладені на них функції. Для VM не має великого значення, на якому саме фізичному сервері вона розгорнута і працює. Строго кажучи, в архітектуру NFV також необхідно включити підсистему підтримки операцій і бізнесу (OSS / BSS), яка є частиною традиційної системи оператора зв'язку. Однак, наявність цієї підсистеми в архітектурі NFV є тимчасовим, оскільки оператори не можуть

одномоментно відмовитися від існуючих OSS / BSS і відразу перейти на MA- (таке можливо тільки для нових мереж операторів). MA- повинна мати повну видимість (операційне стан, статистика використання та ін.) всіх програмних сутностей, розгорнутих в системі NFV, і управляти ними. Тому саме MA- представляє найбільш підходящий інтерфейс для підсистеми OSS / BSS в частині збору операційних даних. В майбутньому, у міру трансформації мережі, всі функції OSS / BSS повинні перейти до MA.

Висновки

У цьому розділі були введені ключові поняття, необхідні для розуміння типової архітектури та роботи SDN, таких як поділ площин управління та даних, а також те, як контролери контролюють та виявляють мережеві пристрої.

Також була представлена узагальнена архітектура NFV яка складається з віртуалізуваних мережевих функцій, інфраструктури віртуалізації та підсистеми управління та оркестрації.

РОЗДІЛ 2
МЕТОДИ ВІРТУАЛІЗАЦІЇ

2.4 Традиційні методи віртуалізації

2.4.1 VLAN

В даний час технологія віртуальної локальної мережі (VLAN) стандартизована стандартом IEEE 802.1Q-2014 [24]. VLAN забезпечують надійну ізоляцію трафіку та зменшення трансляції «бродкаст» доменів у локальних мережах, і це одна з найбільш основних форм віртуалізації мережі. Основна функціональність реалізована шляхом введення тегу VLAN (тег 802.1Q) в середину кадрів Ethernet. Наприклад, комутатори, які підтримують VLAN не дозволяють пересилати кадри, позначені певним тегом VLAN X до портів, які належать лише VLAN Y. До полів в кадрі рівня 2 Ethernet II, що підтримує VLAN, входять [24]:

- Призначення MAC (6 байтів)
- Джерело MAC (6 байт)
- Тег VLAN (4 байти)
- Етертип (2 байти)
- Корисне навантаження (змінної довжини)
- Перевірка CRC (4 байти)

4 байти тегу VLAN розділяються наступним чином [24]:

- Ідентифікатор протоколу тегу (TPID, 2 байти)

  - 0x8100 для базової функціональності VLAN.

  - 0x88a8 для адресації основних компонентів (Q-in-Q).

  - 0x88e7 для інкапсуляції послуг (MAC-in-MAC).

- Інформація про управління тегами (TCI, 2 байти)

- Пріоритетна кодова точка (PCP) - 3-бітове поле, яке можна використовувати для передачі пріоритетної інформації про поточний кадр.

- Припустимий індикатор падіння (DEI) - 1-бітове поле, яке можна використовувати, щоб вказати, чи можна цей кадр відключити у випадку перевантаження.

- Ідентифікатор VLAN (VID) - 12-бітове поле, що використовується для ідентифікації, до якої VLAN належить цей кадр.

Крім того, використання тегів VLAN також дозволяє позначити трафік різними пріоритетами. Наприклад, для часових кадрів, сформованих IP-телефоном, поле PCP може бути налаштовано на більш високе значення, що вказує на те, що цей кадр слід пересилати з більшим пріоритетом у мережі.

Деякі проблеми виникають із використанням фреймів з одним тегом VLAN. Наприклад, можна створити лише 2^{12} віртуальних мереж, число, якого не вистачає для поточних мереж ЦОД з тисячами віртуальних машин. Крім того, кожен клієнт може використовувати лише підмножину з 2^{12} VID. Цю проблему частково вирішує Q-in-Q, що пояснюється в наступному розділі.

2.4.2 Q-in-Q

Використання двох тегів VLAN на кадрі дозволяє клієнтам використовувати повний простір заголовка VLAN, долаючи обмеження VID. Використання двох тегів VLAN в одному кадрі, як правило, постачальники мережевого обладнання називають Q-in-Q [31] і спочатку було вказано стандартом IEEE 802.1ad-2005 [26]. Стандарт описує ізоляцію безлічі клієнтських мереж в межах одного провайдера, присвоюючи один VID кожному клієнту в першому тегу VLAN кадру, і клієнт може вільно використовувати будь-який з 2^{12} VID у другому тегу VLAN. Назва Q-in-Q є посиланням на той факт, що тег 802.1Q знову використовується в рамках 802.1Q. Потім

оригінальний стандарт 802.1ad був включений у стандарт IEEE 802.1Q-2014 [24].

На рисунку 2.4.1 показані різні типи кадрів для ілюстрації та порівняння різних варіантів тегування та інкапсуляції, можливих для VLAN - чистий кадр Ethernet, кадр з одним тегом VLAN - дозволяє 2^{12} різних VID - і кадр з двома VLAN теги - дозволяючи $2^{12}*2^{12} = 2^{24}$ комбінацій VID.

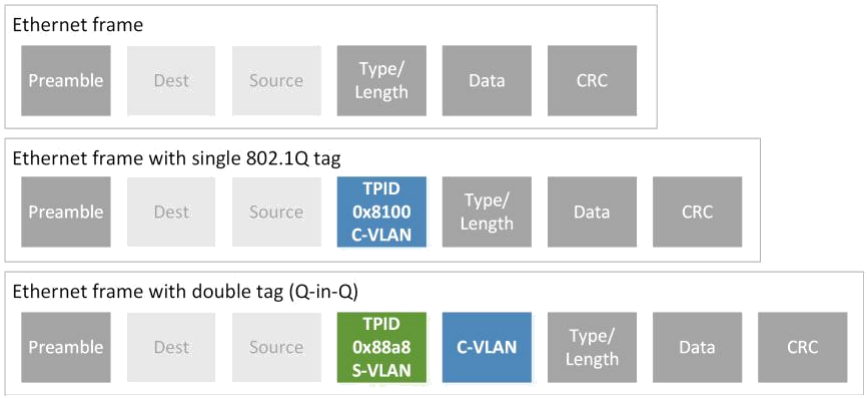


Рис. 2.1 Оригінальний кадр Ethernet та різні варіанти інкапсуляції.

Однак, MAC-адреса замовника завжди відображається в цих кадрах, незалежно від кількості тегів VLAN, що використовуються для корисного навантаження.

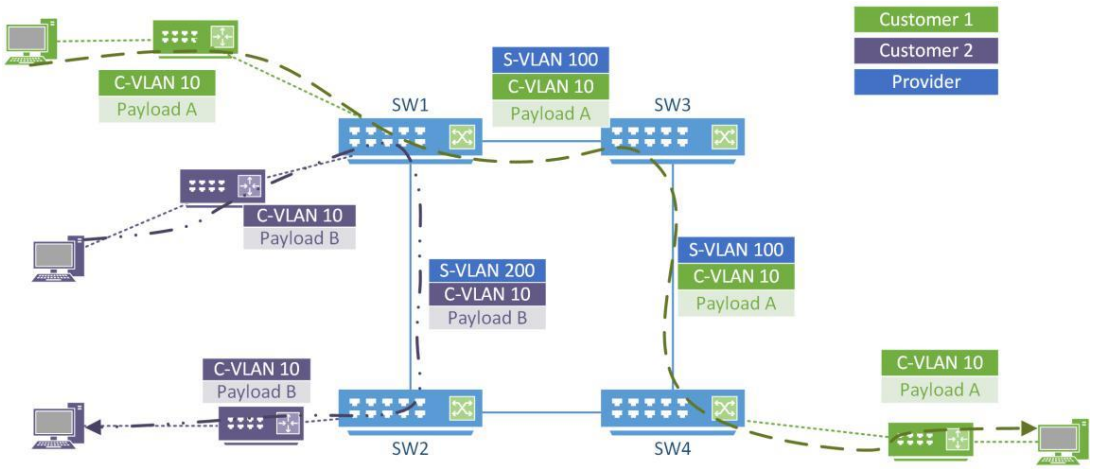


Рис. 2.2 Використання повторюваних тегів VLAN, що входять до Q-in-Q.

На рисунку 2.2, приклад використання Q-in-Q показує двох різних клієнтів, які можуть використовувати один і той же тег VLAN - тег клієнта VLAN (C-VLAN) номер 10 - і залишатись ізольованими один від одного при спільному використанні мережевої інфраструктури. Однак Q-in-Q все ще не забезпечує повного розділення між доменами клієнта та постачальника через такі можливі проблеми [7]:

• MAC-адреси клієнта подорожують по всій магістралі постачальника, і їх запам’ятовують у кожному комутаторі. Це впливає на масштабованість послуги, оскільки таблиці MAC постачальника можуть легко заповнитись;

• Кількість службових тегів VLAN все ще обмежена 2^{12};

• Немає чіткої точки розмежування між мережами споживачів та постачальників щодо управління несправностями та характеристиками.

2.4.3 MAC-in-MAC

Функціонал магістральних мостів провайдера (PBB) постачальники мережевого обладнання зазвичай називають "MAC-in-MAC" який розширює концепцію Q-in-Q, дозволяючи повну інкапсуляцію трафіку клієнта, включаючи MAC-адресу клієнта (C-MAC). Стандарт IEEE 802.1ah-2008 [27] ввів концепцію PBB, яка згодом була включена до стандарту IEEE Std 802.1Q-2014 [24].

Рис.2.3 Кадр MAC-in-MAC.

На рисунку 2.3 показано кадр MAC-in-MAC, в якому знаходяться основні адреси призначення та адреси джерела (B-DA та B-SA відповідно). На рисунку показано, що в кадрі PBB, MAC-адресу клієнта (у світлішому сірому кольорі) не можна дізнатись за допомогою комутаторів на магістралі постачальника. Це

стає зрозумілим при порівнянні кадру PBB з одинарними та подвійними позначеними кадрами на рисунку 2.1. Магістральна VLAN (B-VLAN) являє собою VLAN в магістралі, яка не залежить від інших тегів VLAN у фреймі клієнта. Тег екземпляра серверної служби (I-TAG) містить 24-бітове поле, яке називається ідентифікатором екземпляра серверної служби (I-SID), яке використовується для ідентифікації унікального клієнта в мостовій мережі магістралі постачальника. Це дозволяє 2^{24} різним клієнтам використовувати 2^{24} різних комбінації тегів VLAN кожному [7].

Тому рішення MAC-in-MAC вирішує основні проблеми, які виникають із Q-in-Q, шляхом:

• Надання більшої кількості ідентифікаторів для різних споживачів мережі (224);

• Запобігання запам’ятовування комутаторами на магістралі всіх клієнтських MAC-адрес, вивчаючи лише адреси призначення та адресу відправника;

• Введення чіткої точки розмежування між мережами клієнтів та провайдерів.

2.4.4 Multiprotocol Label Switching

Архітектура багатопротокольної комутації міток (MPLS) вводить концепцію розділення наборів пакетів на класи еквівалентності пересилання (FEC) та зіставлення кожного набору з певним набором хопів у мережі. Потім FEC позначаються мітками або шляхами з комутацією міток (LSP) по мережі. Потім мітки можуть бути легко використані маршрутизаторами комутації міток (LSR) для прийняття рішень щодо переадресації, оскільки мережевий рівень аналізується лише один раз - на мережевому маршрутизаторі [54].

Шляхи, які проходять FEC в мережі, зазвичай призначаються алгоритмами маршрутизації, які вже працюють у мережі, наприклад,

найкоротший шлях спочатку. Потім відображення LSP до FEC зазвичай виконується протоколом розподілу міток (LDP) [4] або протоколом резервування ресурсів (RSVP) з розширеннями для тунелів LSP [6], що дозволяє різним LSR узгоджувати значення міток та способи пересилання їх через мережу.

Крім усього іншого, MPLS дозволяє мережі з комутацією пакетів працювати майже як мережа з комутацією каналів. Багато мережевих служб базуються на MPLS, таких як послуги віртуальних приватних каналів (VPWS) та послуги віртуальної приватної локальної мережі (VPLS) [5].

2.4.6 Virtual Private Networks

Віртуальні приватні мережі (VPN) досягають віртуалізації таким чином, що мережевий клієнт зазвичай бачить лише частину всієї мережі, якою може бути Інтернет або будь-яка інша спільна мережева інфраструктура. Тільки користувачі, які знаходяться в одній мережі, можуть надсилати та отримувати трафік у мережі VPN [34].

Хоча VPN часто вважають віртуалізацією мережі з доданими механізмами безпеки, такими як шифрування та аутентифікація, немає точного визначення того, що VPN повинна включати. У деяких випадках шифрування та аутентифікація взагалі не використовуються (наприклад, провайдеру мережі довіряють MPLS VPLS VPN). Льюїс [34] та Джаха [30] спробували класифікувати деякі з існуючих типів VPN різними способами. Деякі з цих критеріїв узагальнені в наступних параграфах.

У надійних мережах VPN клієнт вірить, що мережа постачальника є безпечною та недоступною для громадськості. Не потрібна аутентифікація чи шифрування. Прикладами надійних VPN є VPN рівня 2 на базі MPLS, такі як віртуальна служба приватних каналів (VPWS) та служба віртуальної приватної локальної мережі (VPLS) [5]. У захищених VPN клієнтські дані повинні бути

аутентифіковані та зашифровані через мережу провайдера. Прикладами надійних VPN є рівень захищених сокетів (SSL) на основі додатків (наприклад, OpenVPN) , та безпека інтернет-протоколів (IPSec) [32].

VPN, орієнтовані на з'єднання, використовують віртуальні схеми або тунелі для передачі даних. Наприклад, VPN рівня 2 MPLS або загальна інкапсуляція маршрутизації (GRE) являється орієнтованим на з'єднання [18]. VPN без підключення покладаються на розподіл даних клієнта на межі провайдера (PE) [30], [34]. Використання тегів VLAN для досягнення зв’язку другого рівня між двома клієнтськими сайтами можна вважати беззв’язковим, оскільки в основному для пересилання він покладається на «чистий» Ethernet [23].

Мережі VPN, що надаються постачальником, повинні бути повністю сконфігуровані та розгорнуті провайдером мережі. Це дозволяє постачальнику точніше контролювати, як обробляється трафік клієнта. VPWS, VPLS та VRF - це всі VPN-мережі, що надаються постачальником послуг [30], [34]. VPN що забезпечені клієнтом конфігуруються та розгортаються повністю мережею клієнта, а постачальник може навіть не знати про існування цих VPN. GRE та SSL, засновані на додатках VPN, що забезпечені клієнтами [30], [34].

VPN може забезпечити ті самі переваги віртуалізації, що були розглянуті раніше, топологія, адресний простір та ізоляція ресурсів. У наступних кількох розділах буде представлено підхід SDN до віртуалізації мережі - мережеві гіпервізори.

2.5 Віртуалізація в мережах SDN

2.5.1 FlowVisor

FlowVisor [60] - один із перших підходів до віртуалізації мереж SDN, який був реалізований на основі OpenFlow. Він складається з прозорого проксі-сервера, який діє між контролером OpenFlow і комутаторами, дозволяючи

декільком контролерам спільно використовувати одну і ту ж мережеву інфраструктуру.

Щоб ізолювати топологію, FlowVisor може обмежити порти, які бачать контролери користувача в реальній фізичній топології, щоб приховати обмежені ділянки мережі від користувачів [60]. На рисунку 2.4 показаний приклад, коли лише Користувач 1 бачить комутатори SW1 і SW2 з усієї мережі, а Користувач 2 бачить лише частину портів від комутаторів SW3 і SW4.

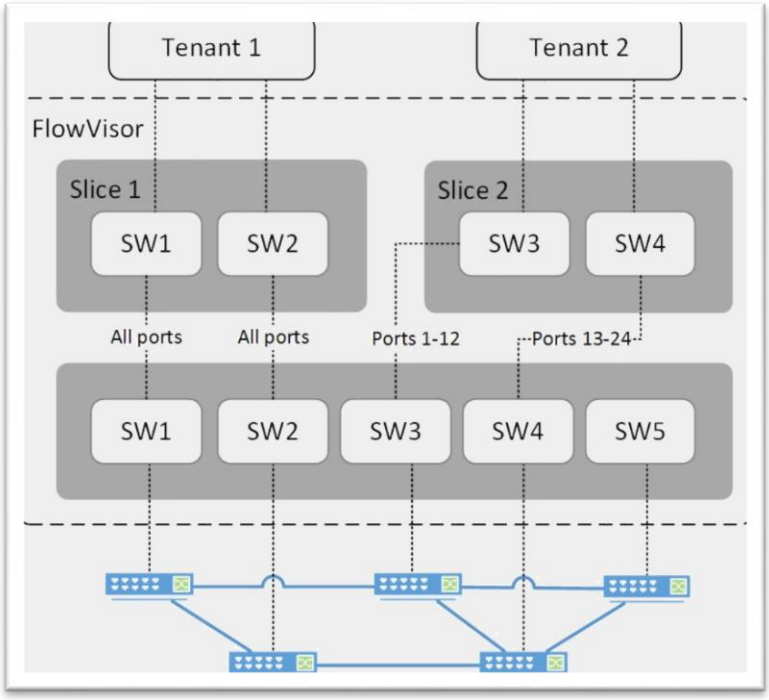


Рис.2.4 Ізоляція мережі за допомогою FlowVisor.

Ізоляція потокового простору реалізується шляхом переписування правил потоку, надісланих від контролера користувача на комутатор, обмежуючи потік, на який вони впливають. Наприклад, якщо контролер намагається створити правило, що обмежує весь трафік, і FlowVisor знає, що цей контролер має доступ лише для встановлення правил для TCP-порту 80, тоді OpenFlow правило перезаписується у FlowVisor перед надсиланням комутаторам і впливає лише на TCP-порт 80 трафіку [60].

Щоб захистити центральний процесор мережевих пристроїв, FlowVisor діє обмежуючи швидкість повідомлень, якими обмінюються контролери користувача та комутатори. Якщо із мережевого пристрою на контролер надходить занадто багато повідомлень про пропущення таблиці OpenFlow PACKET IN, FlowVisor також діє шляхом встановлення правила тимчасового падіння швидкості у відповідному мережевому пристрої під час пересилання першого пропуску таблиці PACKET IN на контролер. Це захищає не тільки центральний процесор мережевого пристрою, але і центральний процесор контролера, оскільки дає час контролеру обробити новий запит про пропуск таблиці PACKET IN, не заважаючи подальшим повторним повідомленням PACKET IN, що надходять з тієї ж мережі пристрій [60].

FlowVisor також забезпечує захист таблиць потоків вимикачів, щоб гарантувати, що один контролер не вичерпує всі записи потоку в пристрої. Це досягається шляхом підрахунку правил потоку, встановлених кожним контролером користувача, та обмеження кількості дозволених правил потоку до заздалегідь визначеного значення [60]. Щоб обмежити пропускну здатність мережі, OpenFlow не підтримує прямий спосіб управління пропускною здатністю або QoS. Тому він може використовувати для цього різні мітки VLAN, сконфігуровані з різними бітами PCP.

SW1 пересилає цей запит FlowVisor всередині повідомлення OpenFlow PACKET IN. На кроці 1 FlowVisor аналізує повідомлення PACKET IN та передає цей запит ARP контролеру A (крок 2), оскільки MAC-адреса хосту A1 належить простору потоку фрагмента A (це налаштовано адміністратором мережі в базі даних FlowVisor). Контролер, що запускає просту програму MAC Learning, надсилає FlowVisor потік PACKET OUT із запитом ARP (Крок 3). При необхідності FlowVisor переписує повідомлення PACKET OUT перед тим, як відправити його назад у мережу (Крок 4), гарантуючи, що ARP-запит буде

заповнено лише для членів зрізу A. SW1 отримав PACKET OUT, пересилає запит ARP на SW2, який, у свою чергу, повторює ті самі кроки 1-4, щоб переслати запит ARP на хост A (крок 5). A2 отримує запит ARP, відповідає на нього, і починається зворотний процес досягнення A1. Коли контролер A отримує повідомлення відповіді "PACKET IN ARP" від A2, контролер має достатньо інформації, щоб наказати комутаторам розпочати переадресацію руху між A1 і A2 незалежно (Крок 6). На кроці 7 контролер A надсилає повідомлення FLOW MOD FlowVisor. FlowVisor перевіряє, чи не порушують ці повідомлення жодного простору потоку, що не належить контролеру A, а потім програмує SW1 та SW2 за новими правилами потоку. Крім того, контролер A закінчує надсилання відповіді PACKET OUT ARP назад на SW1, щоб переконатися, що завершується фаза роздільної адреси. Програмування апаратної частина SW1 і SW2 виконано, і хости A1 і A2 можуть обмінюватися даними через площину даних без подальшої взаємодії з площини управління. Етапи 8-10 показують, як FlowVisor обробляє випадок, коли і хост, і контролер користувача намагаються «вторгнутися» в частину простору потоку, яка не належить їх фрагменту. FlowVisor блокує спроби контролера B встановити потоки в зрізі контролера A.

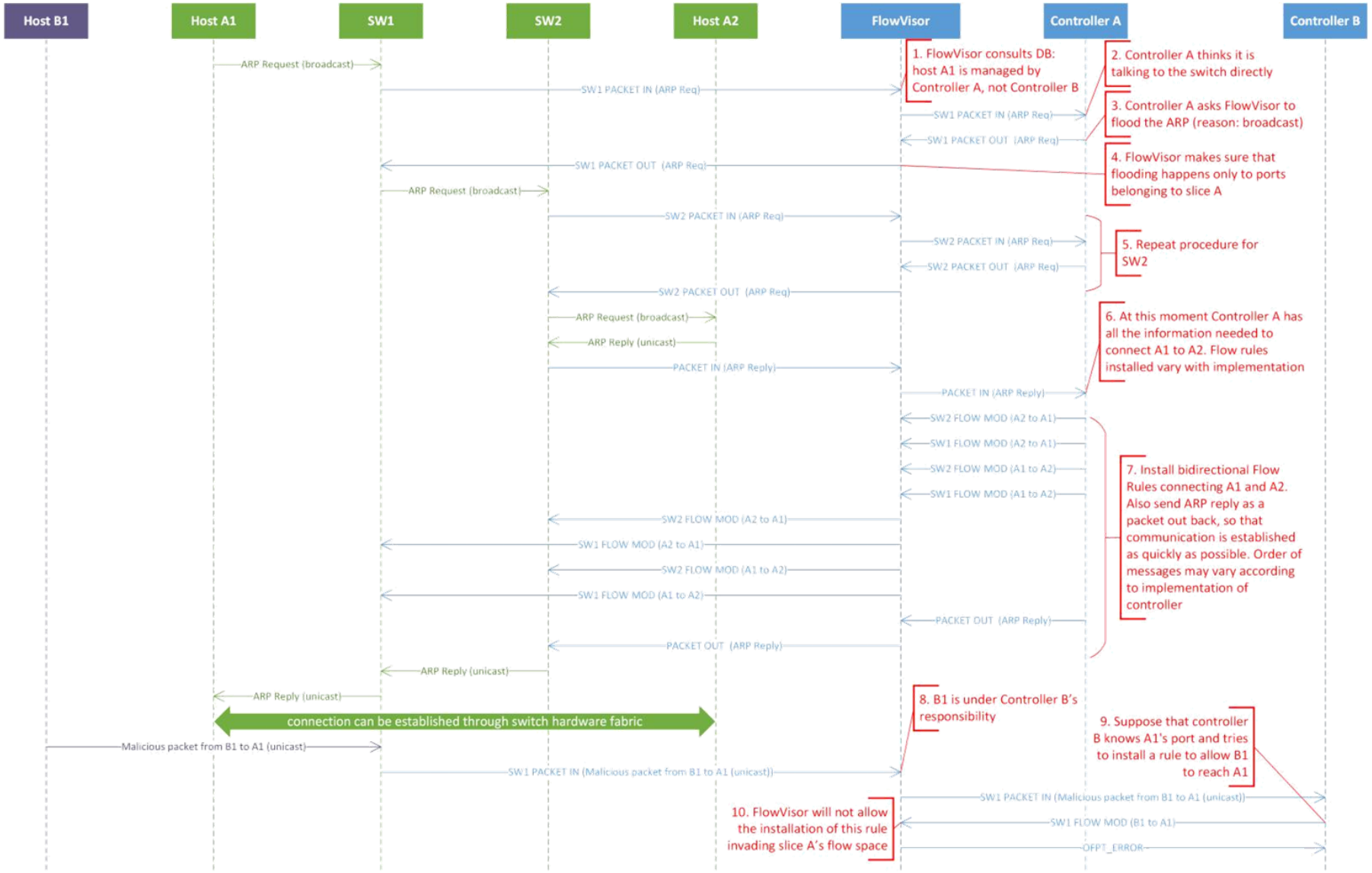


Рис.2.5 Обробка потоку FlowVisor.

2.5.2 VeRTIGO

VeRTIGO [15] розширює FlowVisor концепцією абстрактних вузлів або абстрактних мережевих пристроїв, які складаються з двох основних блоків: віртуальних посилань та віртуальних портів. Віртуальні посилання об’єднують кілька фізичних вузлів та посилань, абстрагують їх до контролера оренди як єдине посилання між будь-якими двома портами. Один або кілька віртуальних портів відображаються фізичний порт відповідно до кількості віртуальних посилань, які повинні використовувати той самий фізичний порт. На рисунку 2.6 показано, як VeRTIGO використовує віртуальні посилання 1 і 2 між SW-A і SW-D для створення надмірних з'єднань між фізичними портами A і B. Чотири фізичні комутатори представлені у вигляді єдиного абстрактного вузла для контролера оренди з віртуальними портами X та Y, що зіставляються з фізичними портами A та B відповідно.

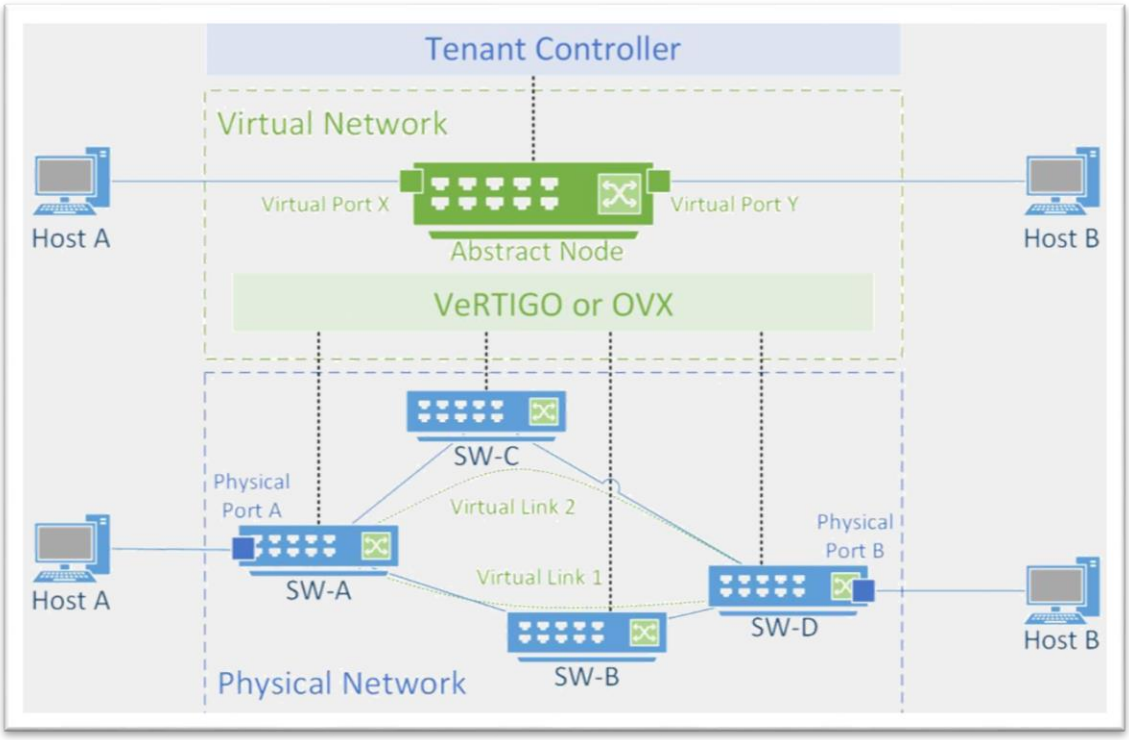


Рис. 2.6 Фізична мережа внизу та відповідна віртуальна мережа.

VeRTIGO побудований поверх FlowVisor, тому він автоматично надає всі функції розділення мережі, що надаються FlowVisor. Тут коротко описані нові модулі, розроблені спеціально для VeRTIGO.

Модуль класифікатора визначає, які повідомлення, що надходять з мережі, обробляються контролерами орендаря OpenFlow. Деякі повідомлення обробляються безпосередньо внутрішнім контролером VeRTIGO, щоб він міг приховати деталі мережі від контролерів користувачів, які контролюють лише частину. Наприклад, на рисунку 2.6 повідомлення OpenFlow, генеровані шляхом руху трафіку в SW-A з віртуального порту X, обробляються контролером користувача, тоді як будь-які повідомлення OpenFlow, пов’язані з трафіком між SW-B та SW-D - внутрішня частина віртуальної мережі , приховані VeRTIGO - обробляються та маршрутизуються внутрішнім контролером [15].

Коли один фізичний порт використовується багатьма різними віртуальними посиланнями, то портувальник портів обробляє відображення віртуальних до фізичних портів. Це також відбувається для портів абстрактних вузлів, які повинні мати фізичні номери портів, переназначені до номерів віртуальних портів [15].

Планувальник віртуальної топології (VT) відповідає за асоціювання екземплярів віртуальної мережі з реальними мережевими ресурсами. Наприклад, коли існує кілька шляхів для з'єднання точок A і B, цей модуль відповідає за пошук найкращого шляху всередині абстрактного вузла. Найкращий шлях залежить від вимог програми і може бути обраний на основі доступної пропускної здатності або загальної затримки. Планувальник VT контролює мережну статистику для визначення поточної пропускної здатності та затримки, щоб визначити, який найкращий шлях для поточної програми.

2.5.3 OpenVirteX

OpenVirteX (OVX) – забезпечує структуру, яка дозволяє контролерам користувачів створювати екземпляри, знімки, мігрувати або видаляти віртуальні мережі, що є аналогом гіпервізорів, що обробляють віртуальні машини (ВМ) в середовищах хмарних обчислень. Подібно до FlowVisor, OVX виступає як проксі-сервер між мережевою операційною системою (-S) та мережею, що підтримує OpenFlow. Основними вдосконаленнями, що він забезпечує є:

• Повний спеціальний адресний простір (або простір потоку) для кожного з створених фрагментів мережі без ризику перекриття адрес;

• Повністю віртуалізована топологія мережі, яку може вказати користувач.

Зразок віртуалізованої мережі великих комутаторів, показаний на рисунку 2.6., також може бути реалізований за допомогою OVX, з різницею, що OVX може забезпечити майже повний простір заголовків MAC та IPv4 кожному з своїх орендарів. Це потенційно дозволяє орендарям використовувати адреси які перекриваються. Це досягається за допомогою методів перезапису заголовків на межах мережі, переписуючи MAC або IP-адреси в заголовки пакетів з додатковою інформацією, включаючи глобально-унікальні ідентифікатори користувача. Якщо контролер користувача передає правила потоку рівня 2 на мережеві пристрої, то OVX вдається до перезапису заголовка MAC. Якщо встановлені правила потоку рівня 3, OVX використовує перезапис IP [57].

Аль-Шабібі та ін. [57] заявляють, що віртуалізація посилань також може бути реалізована за допомогою міток MPLS замість переписування заголовка, хоча це не робиться в поточній версії OVX.

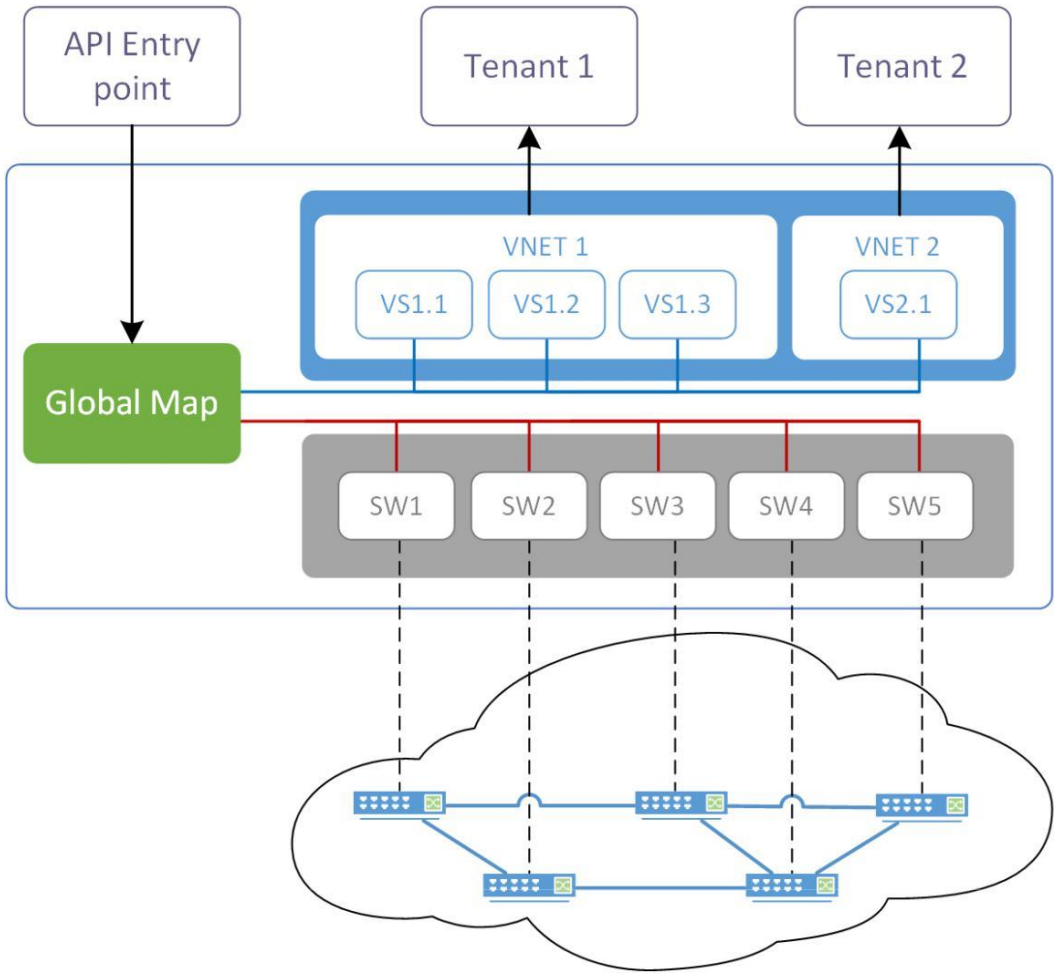


Рис. 2.7 Спрощена архітектура OVX.

OVX працює, зберігаючи дві окремі логічні представлення мережі, як показано на малюнку 11 [45]. Одним із них є фізична мережа, ілюстрована комутаторами всередині хмари, та їх віртуальні аналоги, що зберігаються в базі даних OVX - SW1-SW5 у сірому полі. Віртуальні мережі представлені синіми полями у верхній частині архітектури. Як фізичне, так і віртуальне представлення мережі зберігаються на глобальній карті, представленій зеленим полем на Рис. 2.7, яке в даний час використовує MongoDB як серверну базу даних [46].

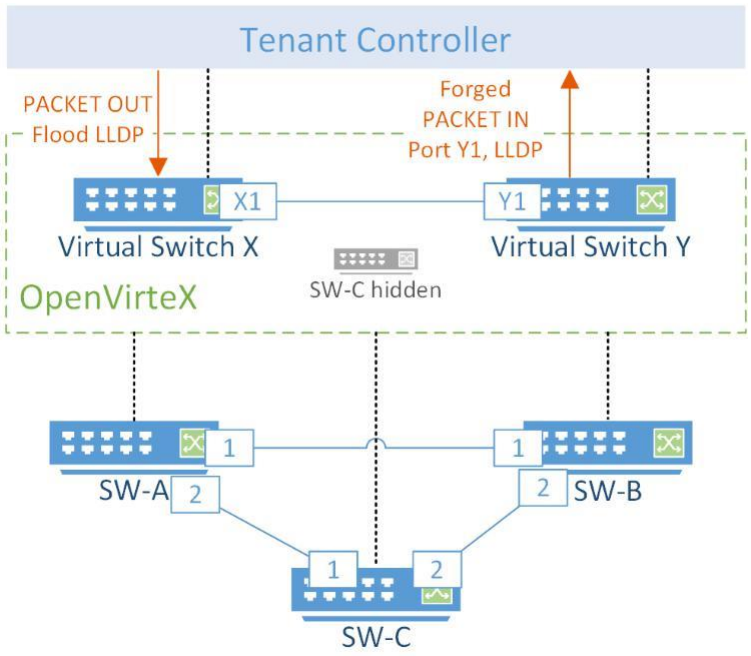


Рис.2.8 Віртуалізація топології OVX за допомогою маніпуляції виявленням.

Щоб забезпечити представлення різних віртуальних топологій, OVX перехоплює повідомлення LLDP, що надходять від контролерів користувача, і створює відповіді LLDP, призначені для представлення до основної топології фізичної мережі різними способами. Ця маніпуляція з виявленням топології зображена на рисунку 2.8, де віртуальний комутатор X зіставляється з комутатором A, віртуальний комутатор Y зіставляється з комутатором B, а комутатор C прихований від контролера користувача. OVX отримує PACKET OUT від контролера користувача, вимагаючи залити кадри LLDP, щоб виявити основну топологію. Замість того, щоб відправити кадр LLDP через порти 1 і 2 SW-A, OVX створює підроблений кадр LLDP як відповідь, щоб надати контролеру користувача ілюзію, що є лише два комутатори, X та Y, підключені між собою через порти X1 та Y1 відповідно.

Додатковою функцією, яку забезпечує OVX, є стійкість завдяки використанню резервних маршрутів у мережі. При побудові абстрактних вузлів з кількома шляхами між хостами OVX може планувати прямі та резервні

маршрути. Якщо посилання в межах прямого маршруту падає, OVX виявляє це і автоматично встановлює правила резервного маршруту в комутаторах [57].

На рисунку 2.9 показано, як OVX працює для забезпечення віртуалізації мережі. З OVX два комутатора SW1 і SW2 представлені у вигляді єдиного великого комутатора (BIGSW) для контролерів користувачів. На кроці 1 OVX звертається до своєї бази даних (попередньо налаштованої адміністратором), щоб визначити, які хости належать до яких віртуальних мереж. Потім OVX пересилає повідомлення PACKET IN контролеру оренди A на основі цієї бази даних. На кроці 2 OVX використовує буфери для локального зберігання повідомлень PACKET IN, таким чином уникаючи надсилання непотрібних даних про корисне навантаження контролерам користувача. Ідентифікатор буфера використовується для ідентифікації буферів, щоб ті самі дані могли бути повернуті в мережу, як показано на кроці 3. На кроці 4, на відміну від FlowVisor, повідомлення PACKET OUT негайно доставляється в SW2, а не з SW1 на SW2 і лише потім доставляється на хост A2. Потім OVX заповнює запит ARP усім хостам, що належать до віртуальної мережі A. Після отримання другого PACKET IN, що відповідає повідомленню ARP-відповіді від A2 до A1, OVX передає його контролеру A. У цей момент (крок 7) контролер A відправляє два повідомлення FLOW MOD для налаштування великого комутатора для з'єднання хостів A1 і A2 . OVX переводить два повідомлення FLOW MOD у чотири повідомлення FLOW MOD для програмування SW1 та SW2. Цей крок є важливим для роботи реалізації великого комутатора, оскільки контролер A бачить і програмує мережу так, ніби це єдиний комутатор. Також на цьому кроці важливо зазначити, що правила FLOW MOD не тільки пересилають трафік, але також переписують IP-адреси джерела та призначення на межах великої комутаційної мережі.

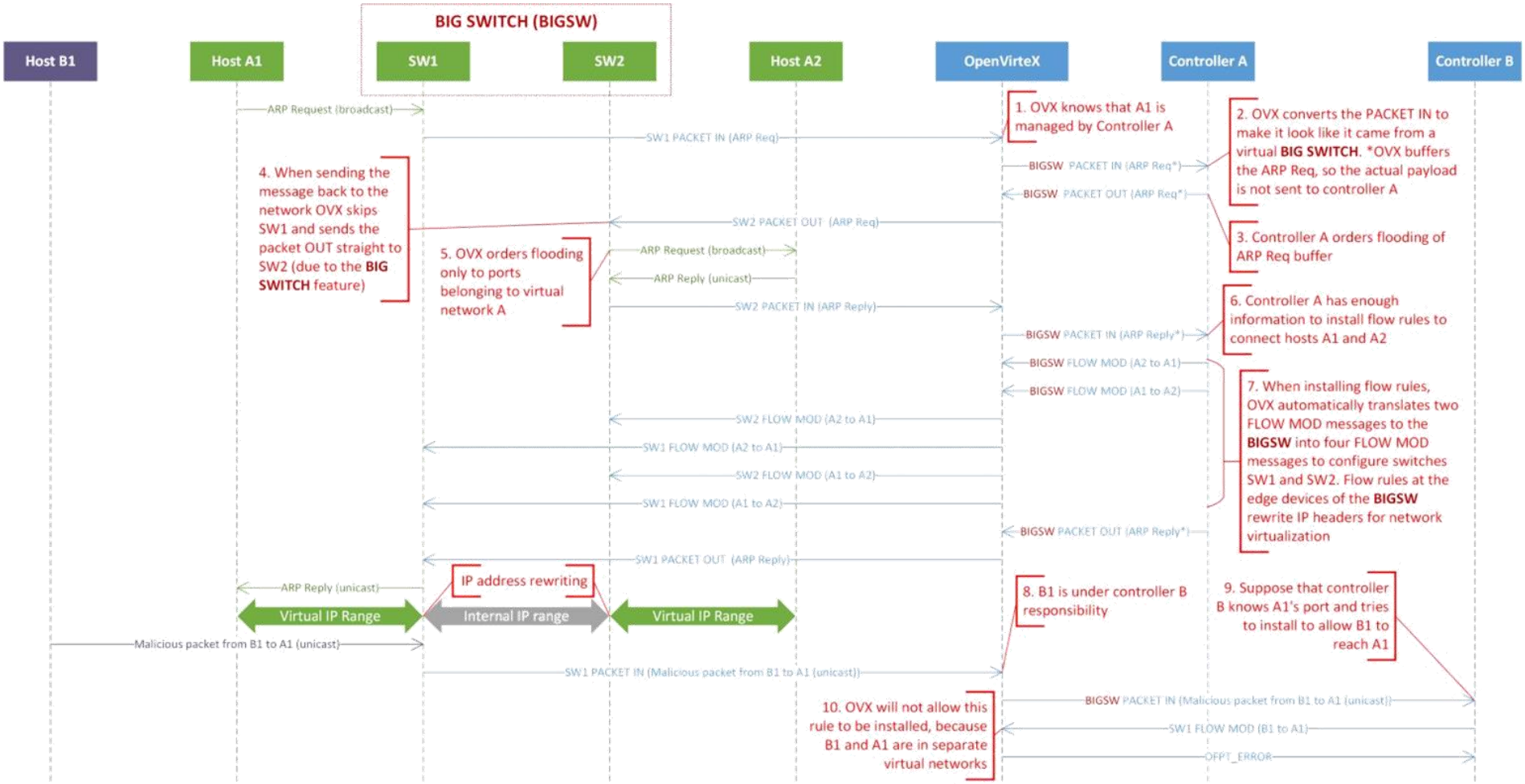


Рисунок 2.9 Обробка потоку пакетів OVX.

2.5.4. FlowN

FlowN [16] пропонує базу даних та полегшену віртуалізацію на основі контейнера як розширення до контролера -X [39]. База даних підтримує відображення фізичної та віртуальної мережі, тоді як кожен віртуальний контейнер запускає одну програму-орендаря з незалежним адресним простором. Щоб забезпечити ізоляцію трафіку, FlowN додає заголовки VLAN для трафіку при вході в мережу і видаляє ці заголовки, коли трафік залишає мережу, дозволяючи орендарям повторно використовувати той самий простір IP-адрес кілька разів.

Замість того, щоб працювати як проксі-контролер, FlowN - це модифікована версія контролера -X. Таким чином, йому не потрібно відображати повідомлення протоколу OpenFlow між фізичною мережею та віртуальним представленням мережі. Це суттєво зменшує накладні витрати на пам'ять декількох контролерів, оскільки віртуалізація реалізується в самому -X, тоді як різні орендарі - це програми, що працюють на одному контролері, але в контейнерах простору імен. Автори FlowN порівнюють його з FlowVisor і показують, що FlowN має чудову продуктивність за наявності великої кількості віртуальних мереж (100 або більше). Покращені результати пояснюються використанням реляційної бази даних для зберігання відображення віртуальної та фізичної топології.

2.5.5. AutoSlice

AutoSlice [9] представляє систему, яка автоматизує завдання розподілу площини пересилання даних мережі між кількома орендарями в площині управління. Його головна мета - мінімізувати ручну реконфігурацію мережі, щоб постачальники субстратів могли перепродавати зрізи своїх мереж орендарям. Реалізація складається з розподіленої архітектури гіпервізора, що складається з одного модуля управління (MM) та декількох проксі-серверів контролера (CPX).

MM відображає топології віртуальної SDN (vSDN) у фізичну мережу та призначає кожному CPX домен мережевих ресурсів. CPX відповідає за перезапис керуючих повідомлень з контрольної мережі на площину пересилання даних, використовуючи теги трафіку, де це необхідно для забезпечення ізоляції. AutoSlice також намагається використовувати такі властивості, як «потоки миші та слона», щоб оптимізувати кешування записів потоку.

2.5.6. AutoVFlow

AutoVFlow [66] працює аналогічно AutoSlice, але пропонує надати більше контролю адміністраторам кожної vSDN, перекладаючи відповідальність за обробку віртуалізації простору потоку на адміністратора кожної мережі. AutoVFlow також стверджує, що застосовує техніку перезапису MAC на межах мережі, щоб гарантувати, що кожен простір заголовка доступний кожному з орендарів.

Висновки

Цей розділ коротко представив і пояснив VLAN, MPLS та VXLAN та VPN, і було показано, що деякі гіпервізори мережі SDN все ще можуть використовувати деякі з цих традиційних технологій для реалізації віртуалізації мережі. Також були представлені методи віртуалізації мережевих функцій в програмно-керованих мережах. У наступному розділі будуть запропоновані експерименти для оцінки деяких з цих гіпервізорів мережі

РОЗДІЛ 3
ОПИС ТА ПОСТАНОВКА ЗАДАЧІ

3.1 Топологія

3.1.1 Топологія Mininet

Експерименти проводяться на, віртуальному тестовому стенді на основі Mininet [36] для оцінки функціональних аспектів віртуалізації мережі та фізичному тестовому стенді з одним комутатором для аналізу основних аспектів продуктивності віртуалізації мережі.

У всіх експериментах контролером OpenFlow використовується Floodlight [19], оскільки це простий у налаштуванні контролер і містить усі функції, необхідні для експерименту з тестованими мережами, а саме виявлення топології та «навчання» MAC.

Вихідний код доступні для FlowVisor [1], VeRTIGO [12] та OpenVirteX [2]. Вихідний код AutoSlice, AutoVFlow та FlowN не знайдено у відкритому доступі, отже у цій роботі висвітлено лише перші три підходи.

Mininet - це мережевий емулятор, і він працює, керуючи екземпляром мереж віртуальних хостів, комутаторів та посилань. Віртуальні хости створюються як окремі простори імен у Linux, а віртуальні комутатори зазвичай працюють на Open vSwitches (OVS). Mininet дозволяє користувачу використовувати зовнішній контролер OpenFlow для управління комутаторами OVS з підтримкою OpenFlow.

Топологія на основі Mininet, зображена на рисунку 3.1, використовується для всіх функціональних експериментів. До кожного комутатора підключено 3 хоста, хоча хости, підключені до комутаторів SW2-SW4, не відображаються для ясності. Крім того, DPID і MAC-адреси були скорочені за допомогою подвійних двокрапок для вираження послідовності нулів. Наприклад, DPID SW3 (00: 00: 00: 00: 00: 00: 00: 03) представлений 00 :: 03, а MAC-адреса хоста C1 (02: 00: 00: 00:

03: 01) відображається як 02 :: 03: 01. Схема адресації віртуальних пристроїв, що використовуються в цій мережі, була розроблена для полегшення налаштування, розуміння та запуску експериментів:

• Хости були призначені для трьох різних мереж, A, B та C. Мережі A та B (10.0.0.0/8) перекриваються з метою перевірки переваг ізоляції мережі. Мережа C (3.0.0.0/8) призначена для перевірки можливих конфліктів із методом переписування заголовка, який характерний для OVX;

• Ідентифікатори DPID комутатора відповідають номерам комутатора 00 :: NN. Наприклад, комутатор SW3 (NN = 03) має DPID 00 :: 03, щоб полегшити пошук та аналіз результатів;

• MAC-адреси хосту були налаштовані так, щоб відображати мережу, якій вони належать, і комутатор, до якого вони підключені, дотримуючись шаблону 02 :: XX: YY, де XX - це мережа, а YY - комутатор.

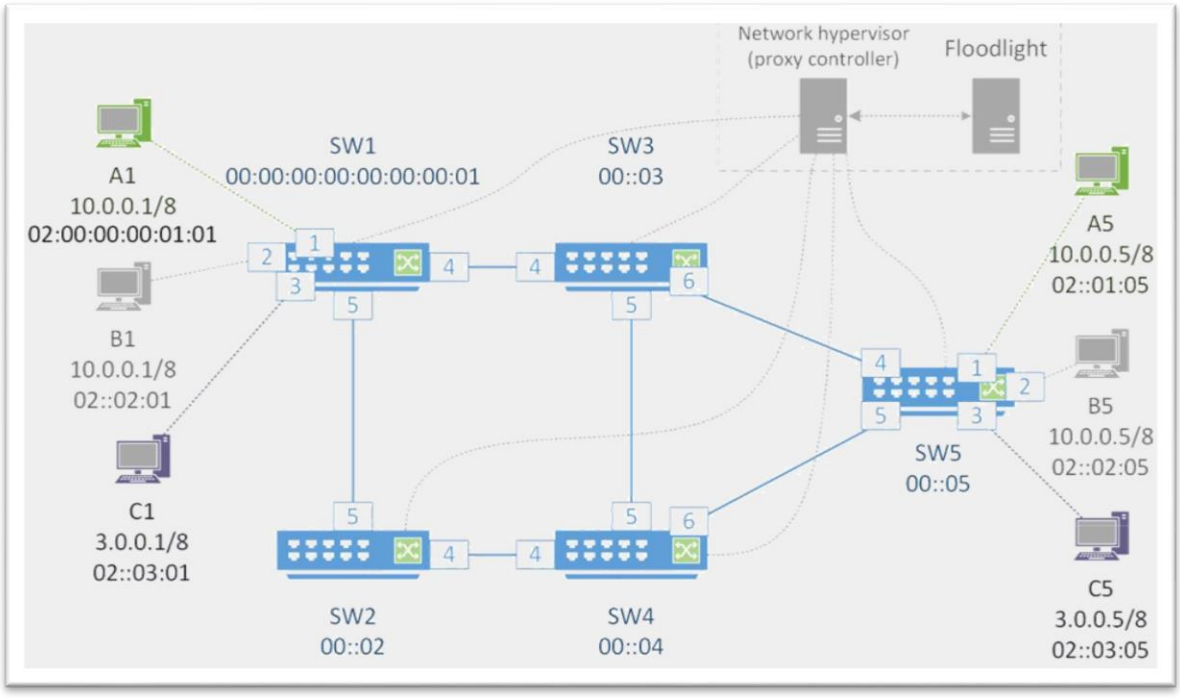


Рис.3.1 Тестовий сценарій Mininet

Рисунок 3.1. також показує, що гіпервізор мережі - OVX, VeRTIGO або FlowVisor підключений до Open vSwitches через інтерфейс зворотного зв'язку. Той самий інтерфейс зворотного зв'язку використовується для підключення мережевого гіпервізора до Floodlight.

Ця мережа може бути використана для демонстрації нарізки FlowVisor, стійкості OVX до резервних маршрутів, а також можливостей віртуалізації топології OVX і VeRTIGO.

3.1.2 Фізична топологія

Фізична топологія, яка використовується для тестування часу налаштування потоку та пропускної здатності між хостами, зображена на рисунку 3.2. Комп’ютери підключені до комутатора HP OpenFlow, який управляється контролером Floodlight, а також будь-яким тестованим гіпервізором мережі - OVX, FlowVisor або VeRTIGO . Тестові комп’ютери від A до D мають два мережеві інтерфейси, кожен з яких підключений до комутатора HP OpenFlow, що використовується в тестах продуктивності, а інший інтерфейс підключений до стандартного комутатора рівня 2 для полегшення управління лабораторним обладнанням. VLAN 10 був налаштований для управління контролером Floodlight. VLAN 40 був налаштований для управління мережевим гіпервізором плюс Floodlight для вимірювання впливу на продуктивність додавання цього рівня віртуалізації мережі до площини управління. Тому комп’ютери A і B використовуються тоді, коли потрібно протестувати лише продуктивність Floodlight, а комп’ютери C і D використовуються, коли тестується мережевий гіпервізор та продуктивність Floodlight.

Для експериментів, проведених у цій локальній фізичній топології, апаратний комутатор OpenFlow підключений до мережевого гіпервізора, який, у свою чергу, підключається до одного екземпляра контролера орендаря Floodlight на порту TCP 10000.

Основною метою цієї фізичної тестової топології є тестування продуктивності, не обмежуючись споживанням ресурсів ЦП Mininet, яких може не вистачати для більш високої пропускної здатності. Це більш проста топологія, оскільки вона призначена лише для вимірювання пропускної здатності трафіку, що перетинає апаратний комутатор, та часу встановлення нових правил потоку в комутаторі.

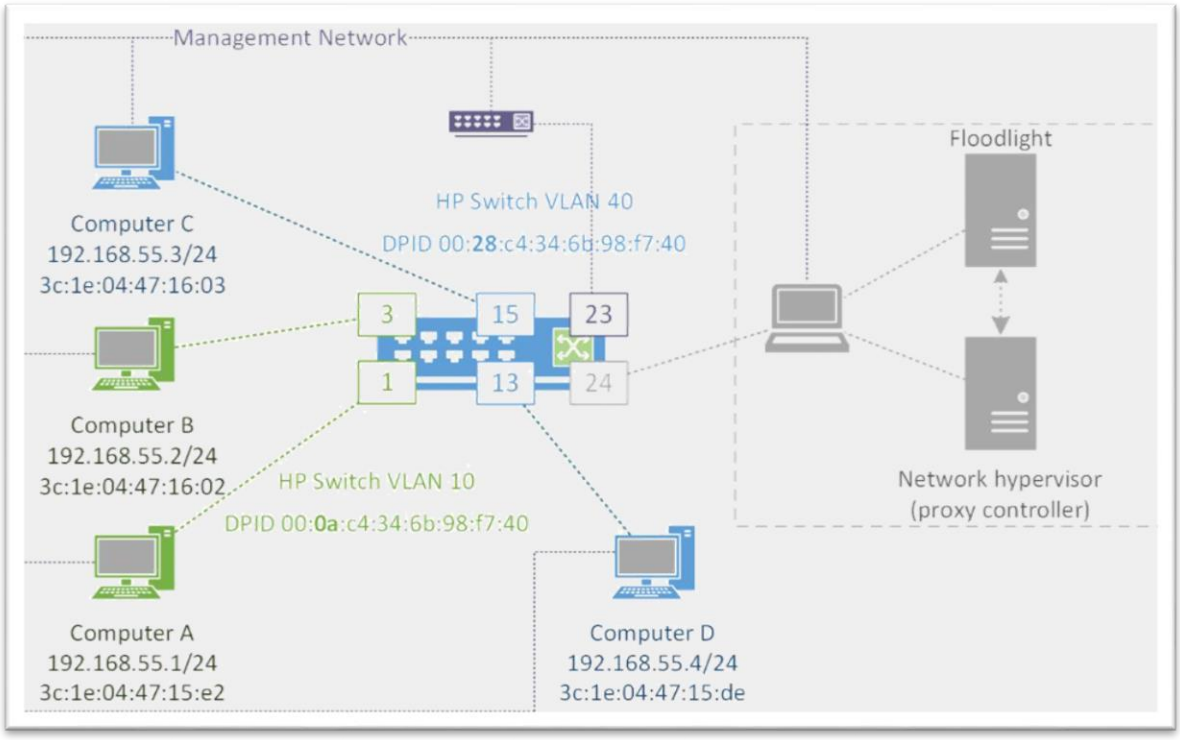


Рис.3.2 Сценарій фізичного випробування для тестів продуктивності

3.2 Ізоляція мережі

Для цього експерименту топологія Mininet використовується для перевірки ізоляції мережі серед орендарів. По-перше, перевіряється мережевий гіпервізор - FlowVisor, OVX або VeRTIGO - повинен бути налаштований на розділення цієї мережі на три окремі віртуальні підмережі, кожна з яких контролюється різним екземпляром Floodlight.

Очікується, що FlowVisor розділить вихідну мережу, як показано на рисунку 3.3. Для OVX та VeRTIGO ізоляція мережі повинна виглядати так, як показано на рисунку 3.2.2., з великим комутатором, що представляє п’ять комутаторів. Ізоляцію мережі можна перевірити, помітивши, що хости, що належать до різних фрагментів, не можуть спілкуватися. Наприклад, хост A1 не повинен мати можливості відправляти будь-які пакети на будь-який з хостів C1-C5. Крім того, мережевий гіпервізор повинен забезпечити, щоб контролери орендарів мали право контролювати лише свої мережеві частини.

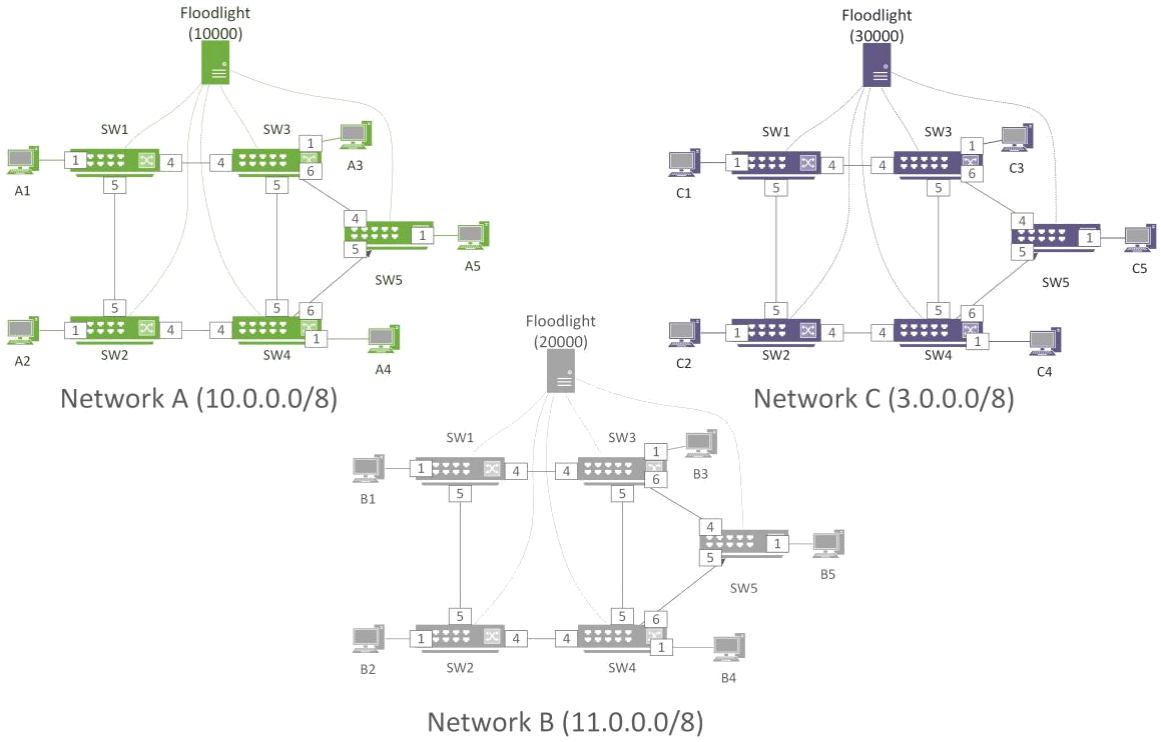


Рис.3.3 Ізоляція топології за допомогою віртуалізації FlowVisor.

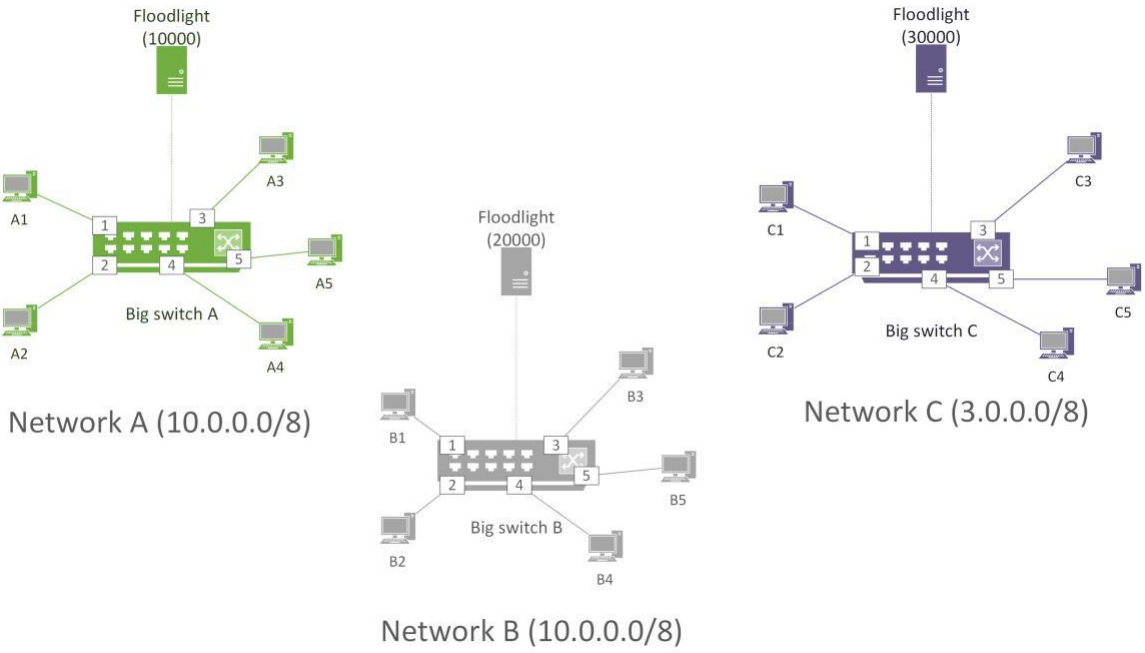


Рис.3.4 Ізоляція топології за допомогою віртуалізації OVX та VeRTIGO.

3.3 Автономне перенаправлення

І OVX, і VeRTIGO стверджують, що можуть забезпечити автономне перенаправлення всередині абстрактних вузлів. Припустимо, що мережевий гіпервізор маршрутизує трафік між хостами A1 і A2 через комутатори SW1 і SW2, як зображено рисунуц 3.3. Якщо зв'язок між SW1 і SW2 не підтримується, мережевий гіпервізор повинен підтримувати функціонування мережі шляхом перенаправлення трафіку через резервний маршрут, можливо, через комутатори SW1-SW3-SW4-SW2.

Час переривання руху може бути виміряно за допомогою генератора руху, такого як iperf. Генератор трафіку повинен бути налаштований на передачу пакетів UDP, щоб переконатися, що рівень програми не повторно надсилатиме втрачені пакети. Відправляючи пакети з відомим розміром і з фіксованою швидкістю, можна виміряти час переривання, підрахувавши кількість втрачених пакетів. Рівняння 3.1 показує, як розрахувати пропускну здатність на основі обсягу отриманих даних ($data_{rx}$) та часу, що минув.

$$\text{bandwidth} = \frac{\text{data}_{rx}}{\text{time}} \tag{3.1}$$

Враховуючи, що data_{rx}= розмір пакета $N_{packets}$ та відомий розмір пакета, пропускну здатність та час, рівняння 3.1 можна переписати як рівняння 3.2.

$$N_{packets} = \frac{\text{time bandwidth}}{\text{packet size}} \tag{3.2}$$

Наприклад, використовуючи розмір пакета 125 байт (1000 біт), пропускну здатність 1 Мбіт/с та надсилання даних протягом 10 секунд повинні призвести до $N_{packets}$=10000 пакетів. Отже, кожен втрачений пакет відповідає $10/1000 = 1$ мілісекунді. Підрахувати скільки часу мережа залишається недоступною, просто підрахувавши кількість втрачених пакетів.

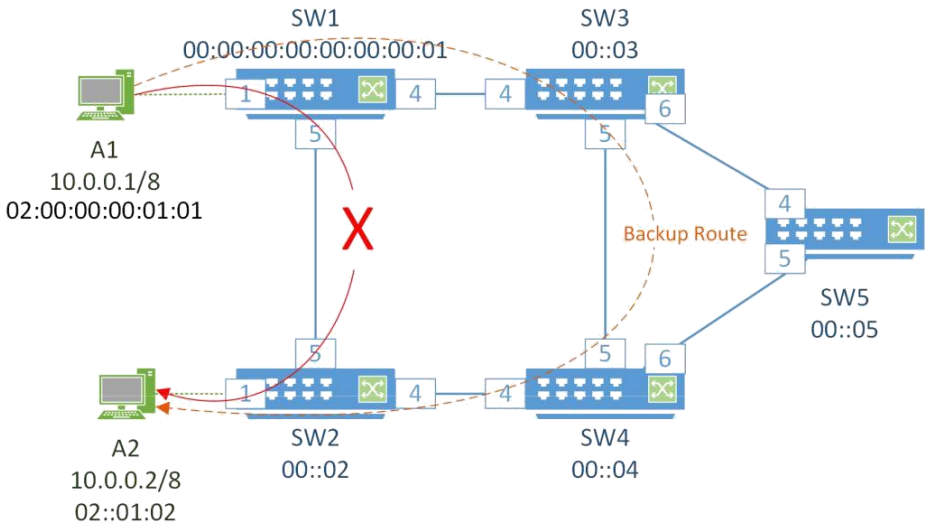


Рис.3.5 Функція швидкого перенаправлення OVX.

3.4 Прозоре пересилання трафіку.

Орендарі мережі можуть захотіти запускати традиційні мережеві механізми у власній віртуальній мережі, такі як механізм виявлення LLDP або протокол маршрутизації початку відкритого найкоротшого шляху (OSPF). Як LLDP, так і OSPF покладаються на використання багатоадресних кадрів для зв'язку з сусідніми мережевими пристроями. Віртуалізація мережі зазвичай добре перевірена для стандартного одноадресного трафіку IP, але деякі багатоадресні кадри та інші спеціальні типи кадрів навряд чи колись перевіряються на прозору переадресацію трафіку в експериментальних мережевих гіпервізорах. Тому важливо перевірити, що віртуальні мережі можуть переадресовувати декілька різних типів трафіку.

Рис.3.6 Еталонне прозоре пересилання трафіку.

Один із способів перевірки прозорості переадресації руху полягає у використанні інструменту test_packets.py [51]. Користувач повинен вводити кадри в один мережевий порт, а потім очікувати, що кадри будуть отримані на будь-якому іншому хості призначення в мережі. Очікується, що мережа переадресовує кадри до портів, що належать до правильного фрагмента або віртуальної мережі, не скидаючи та не змінюючи їх. Різні гіпервізори мережі не повинні додатково обмежувати тип трафіку, який зазвичай може пересилатись лише контролером Floodlight.

3.5 Відповідність стандартам адресації

FlowVisor та VeRTIGO не змінюють трафік у мережі. Вони просто нарізають доступний простір заголовка, щоб різні орендарі обробляли різні контролери орендарів. Однак OVX працює, переписуючи заголовки IP, щоб зробити повний простір заголовка IP доступним для кожного орендаря [57].

Метою цього експерименту є спостереження за тим, як гіпервізор мережі змінює заголовки IP та MAC під час їх надходження в мережу. Трафік, що протікає через внутрішні ланки мережі, фіксується в топології Mininet,потім захоплені кадри аналізуються, маючи на увазі відповідність стандартам адресації IP та MAC та зарезервованим діапазонам адрес, як зазначено:

- IANA IP multicast [29] та розподіл MAC-адрес ;
- розподіл адрес IETF для приватних мереж;
- протоколи управління мостом IEEE, такі як протокол охоплюючого дерева (STP) або загальний протокол реєстрації атрибутів (GARP);
- Інші загальні адреси, вже призначені постачальникам або зарезервовані для конкретних цілей.

Відповідність адресам IPv4 означає, що IP-адреси, сформовані мережевим гіпервізором, є дійсними та знаходяться в межах діапазонів, призначених приватним мережам. Наприклад, 1.0.0.1 - це загальнодоступна IP-адреса, присвоєна організації в Азії [63], а 224.0.0.5 належить до діапазону багатоадресної передачі IP, зарезервованого IANA, і спеціально використовується для обміну повідомленнями OSPF [53]. Жодна з цих адрес не повинна використовуватися мережевими гіпервізорами під час переписування одноадресних заголовків IP, оскільки вони можуть спричинити багато проблем, якщо коли-небудь виникне потреба у взаємодії з іншими IP-мережами.

3.6 Експерименти з продуктивністю

Час налаштування потоку вже визнано одним із недоліків OpenFlow (і SDN), оскільки централізований контролер дбає про пересилання рішень [62]. Крім того, важливо підтвердити, що методи віртуалізації, що застосовуються мережевими гіпервізорами, не впливають суттєво на пропускну здатність мережі. Експерименти, описані в цьому розділі, мають намір вимірювати:

- Накладні витрати часу на налаштування потоку, введені елементом віртуалізації в мережі;
- Зниження пропускної здатності даних, спричинене методами ізоляції руху, такими як переписування заголовків пакетів.

3.6.1 Час налаштування потоку

Завдання цього експерименту полягає у порівнянні латентності встановлених правил потоку Floodlight із затримкою з доданим гіпервізором мережі між Floodlight та комутатором OpenFlow. Цей експеримент проводиться на фізичній топології, описаній у Розділі 3.1.2. Записи ARP на всіх хостах були вручну додані до їхніх таблиць, щоб уникнути додаткової дисперсії, введеної фазою розрішення адреси.

Процедура полягає у вимірюванні часу зворотного зв'язку (RTT) пінгу між двома хостами. Як зображено на рисунку 3.7, коли комутатор не має правил трафіку між хостами A і B, запит exo-відповіді та повідомлення відповіді протоколу керування Інтернетом (ICMP) повинен оброблятися контролером.

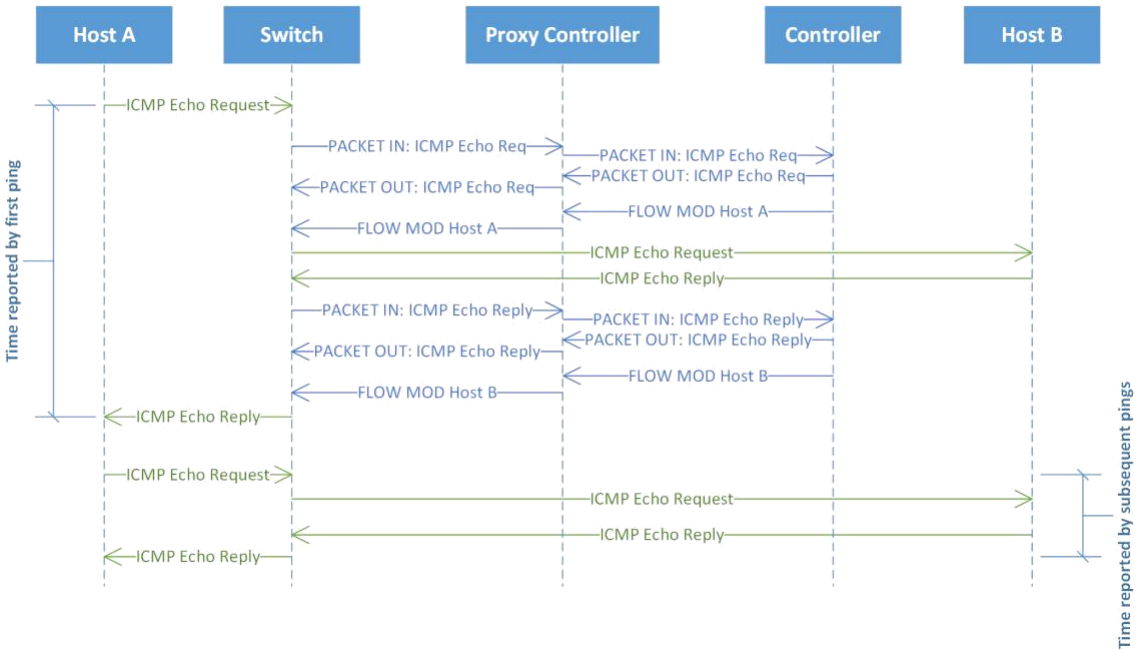


Рис.3.7 Час зворотного зв'язку першого пінгу з мережевим гіпервізором

Порівнюючи RTT першого пінгу з наступними RTT, можна визначити, скільки часу гіпервізор мережі та контролер орендаря додають до всього рівняння RTT 3.3.

$$RTT_{firstping} = t_{processing} + t_{transmission} + t_{queuing} + t_{propagation}$$

(3.3)

Затримки розповсюдження, черги та передачі є незначними в цій невеликій мережі, яка тестується без значного навантаження. Отже, час, необхідний для двостороннього налаштування потоку, залежить майже виключно від часу обробки, доданого проксі-сервером мережевого гіпервізора та контролером Floodlight, як видно з рівняння 3.4:

$$RTT_{firstping} \approx t_{processing}{}_{proxy} + t_{processing}{}_{floodlight} \quad (3.4)$$

3.6.2 Пропускна здатність

Зниження пропускної здатності може бути наслідком впливу методів ізоляції руху, що використовуються у віртуалізації мережі, такими як тегування

VLAN або перезапис заголовків пакетів. Мета цього експерименту - визначити, наскільки знижується пропускна здатність внаслідок віртуалізації мережі.

Один із способів зробити це - визначити максимальну пропускну здатність між двома хостами за допомогою інструмента iperf для кожного випадку.

Очікується, що вплив рівня віртуалізації мережі на пропускну здатність буде незначним. Пропускна здатність, досягнута за допомогою мережевих гіпервізорів, повинна наближатися до пропускної здатності, виміряної при використанні Floodlight окремо.

Висновки

У цьому розділі докладно описані експериментальні установки, що використовуються для оцінки мережевих гіпервізорів, а також деякі інструменти, що використовуються для їх перевірки. Було запропоновано кілька детальних експериментальних процедур. У наступному розділі будуть представлені та проаналізовані результати цих експериментів.

РОЗДІЛ 4

РЕЗУЛЬТАТИ ТА АНАЛІЗ ОТРИМАНИХ ДАНИХ

4.1 Результати функціонального тестування

Функціональні експериментальні результати за своєю суттю якісні і складаються з визначення того, чи є результат прийнятним чи ні. Цей розділ намагається лише представити та коротко проаналізувати результати. Більш глибокий загальний аналіз результатів представлений пізніше в розділі 4.3.

Наступні розділи представляють результати щодо ізоляції мережі та топології; підтримки автономної маршрутизації; прозорої переадресації руху; та відповідність стандартам адресації.

4.1.1 Ізоляція мережі та топології

Графічний інтерфейс Floodlight був використаний для перевірки того, що гіпервізори мережі виставляли правильне подання віртуальної мережі контролерам оренди. Три екземпляри контролера орендаря Floodlight були створені в портах 10000, 20000 та 30000 для управління мережами A, B та C відповідно.

Рисунок 4.1 показує мережу A з точки зору Floodlight, представлену мережевим гіпервізором FlowVisor. Показано п’ять комутаторів з DPID, починаючи з 00 :: 01-05, які відповідають комутаторам SW1-SW5 у топології Mininet, представленій у розділі 3.2.1. Усі хости в цій мережі мають MAC-адреси 02 :: 01: XX, і вони відповідають хостам A1-A5. Аналогічні результати можна спостерігати на рисунках 4.1.2 та 4.1.3, які містять хости B1-B5 та хости C1-C5 відповідно. Це показує, що FlowVisor працює належним чином, дозволяючи ізолювати мережу, розділивши одну мережу на три різні мережі, які працюють незалежно.

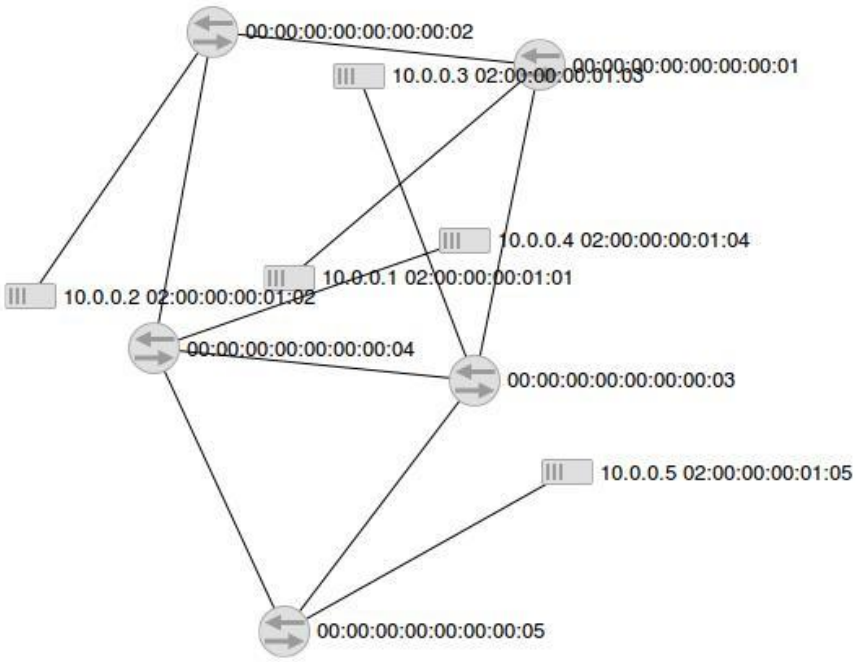


Рис. 4.1 Мережа A, використовуючи FlowVisor як мережний гіпервізор

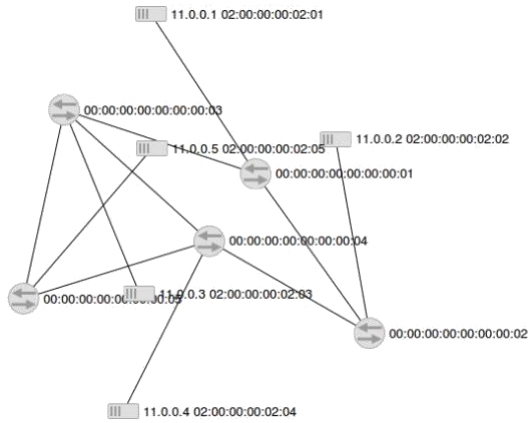


Рис. 4.2 Мережа B

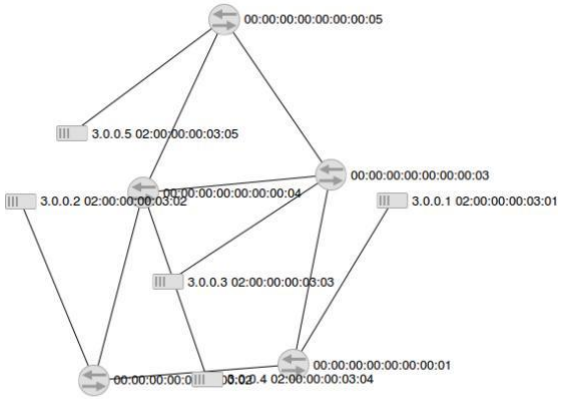


Рис. 4.3 Мережа C

Як показано на рисунку 4.1, Floodlight A «бачить» лише один віртуальний комутатор, представлений OVX. Цей віртуальний комутатор має DPID 00: a4: 05 :: 01 і являється представленням п’яти комутаторів, зображених на рисунку 3.1. Повторюється той самий аналіз, що використовується для FlowVisor, порівнюючи, які хости підключені до кожної мережі. З цього аналізу підтверджено, що хости з MAC-адресою 02 :: 01: XX з'являються лише в мережі

А. Те саме можна підтвердити на рисунку 4.4 та 4.5, де хости 02 :: 02: XX з'являються лише в мережі B і хости 02 :: 03: XX лише в мережі C. Отже, OVX також працює належним чином і дозволяє ізолювати мережу, віртуалізуючи єдину мережу на три абстрактні екземпляри, які працюють незалежно.

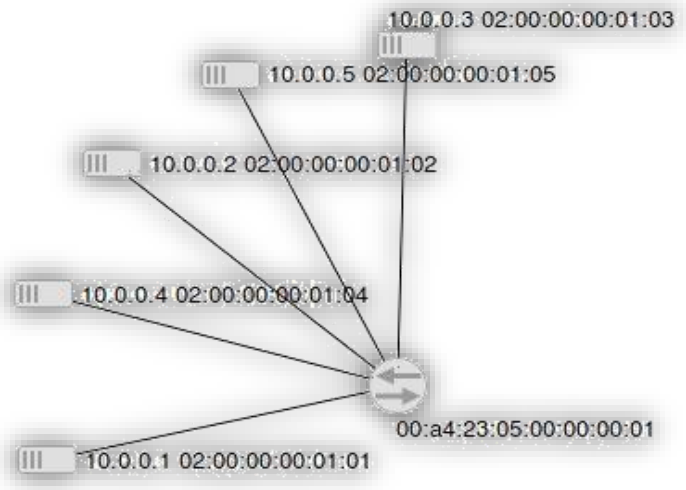


Рис. 4.4 Мережа A з використанням OVX як мережевого гіпервізора.

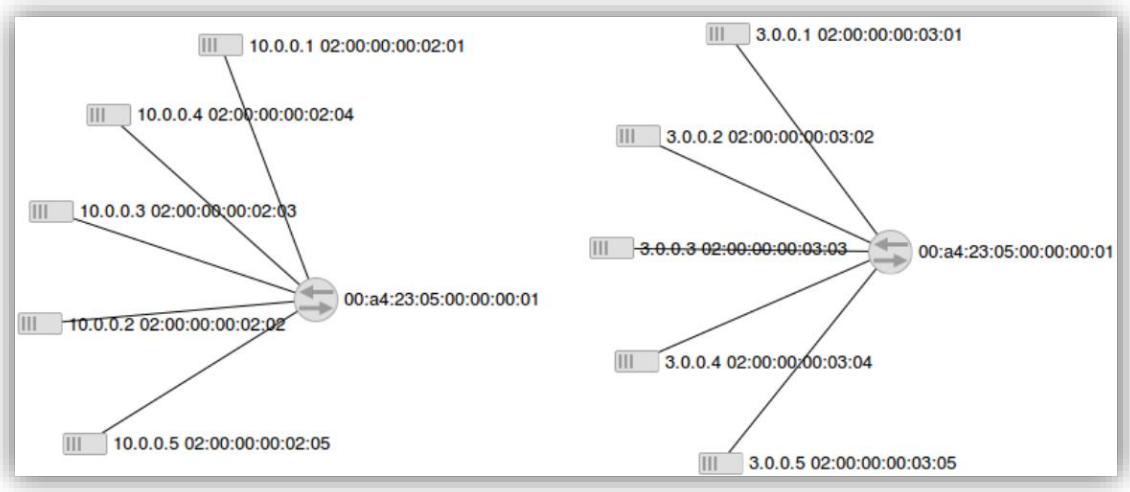


Рис. 4.5 Мережа B з використанням OVX як мережевого гіпервізора

Рис. 4.5 Мережа C з використанням OVX як мережевого гіпервізора

Ізоляція мережі також була перевірена з точки зору мережевих хостів. Використання утиліти pingall від Mininet для всіх хостів показує, що хости A1-A5 не можуть зв’язатися з хостами B1-B5 або C1-C5. Таблиця 4.1 показує цю доступність у матричній формі. Наприклад, перетин рядка B3 зі стовпцем C1 має “N”, що означає, що B3 не може досягти C1. Рядок C5, що перетинається зі стовпцем C1, має "Y", що означає, що C5 може досягти C1. І мережеві гіпервізори OVX і FlowVisor дали правильні результати підключення.

Таблиця 4.1 Підключення хостів, в Mininet

	A1	A2	A3	A4	A5	B1	B2	B3	B4	B5	C1	C2	C3	C4	C5
A1	–	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
A2	Y	–	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
A3	Y	Y	–	Y	Y	N	N	N	N	N	N	N	N	N	N
A4	Y	Y	Y	–	Y	N	N	N	N	N	N	N	N	N	N
A5	Y	Y	Y	Y	–	N	N	N	N	N	N	N	N	N	N
B1	N	N	N	N	N	–	Y	Y	Y	Y	N	N	N	N	N
B2	N	N	N	N	N	Y	–	Y	Y	Y	N	N	N	N	N
B3	N	N	N	N	N	Y	Y	–	Y	Y	N	N	N	N	N
B4	N	N	N	N	N	Y	Y	Y	–	Y	N	N	N	N	N
B5	N	N	N	N	N	Y	Y	Y	Y	–	N	N	N	N	N
C1	N	N	N	N	N	N	N	N	N	N	–	Y	Y	Y	Y
C2	N	N	N	N	N	N	N	N	N	N	Y	–	Y	Y	Y
C3	N	N	N	N	N	N	N	N	N	N	Y	Y	–	Y	Y
C4	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	–	Y
C5	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	–

VeRTIGO не може бути оцінений через проблеми зі стабільністю. Під час висвітлення топології Mininet, VeRTIGO буде перезапускати з'єднання з Floodlight на невизначений час. Здається, проблема полягає у помилці VeRTIGO, яка, швидше за все, пов’язана з інкапсуляцією пакетів LLDP із Open vSwitches. Не вдалося вирішити цю проблему лише за допомогою конфігурації.

4.1.2 Прозоре пересилання трафіку.

За допомогою інструменту test_packets.py було перевірено багато типів трафіку, щоб оцінити, чи блокують контролери віртуалізації певні кадри. Узагальнені результати наведені в таблиці 4.1.2 та класифіковані на шість різних категорій. У таблиці 4.1.2 порівнюються результати пересилання трафіку за допомогою стандартного комутатора рівня 2 (стовпець L2), Open vSwitch (OVS), Floodlight (FL), FlowVisor (FV) та OpenVirteX (OVX). Результати VeRTIGO тут не відображаються, оскільки програма VeRTIGO може натрапити на винятки та перестати працювати під час тестів.

Таблиця 4.2. Узагальнені результати пересилання трафіку.

Type of traffic	L2	OVS	FL	FV	OVX
IPv4 unicast	+	+	+	+	+
IPv6 unicast	+	+	+	-	-
IPv4 multicast	+	+	+	+	+
L2 multicast	+-	-	+-	+-	+-
IPv4 w/ VLAN	+	+	+	+	+
IPv4 w/ MPLS	+	+	+	+	+

І мережевим гіпервізорам OVX і FlowVisor вдалося переслати все, крім:

• пакети IPv6;

• Два типи багатоадресних пакетів L2: протокол виявлення рівня зв'язку (LLDP) і протокол охоплюючого дерева (STP).

Контролер Floodlight підтримував переадресацію IPv6, хоча його потрібно було налаштувати за допомогою OpenFlow 1.3, що не підтримується FlowVisor та OpenVirteX. Однак Floodlight все ще не може пересилати кадри LLDP та STP. Це

обмеження очікується, оскільки ці кадри перехоплюються додатком контролера для цілей виявлення мережі або обробляються / скидаються Open vSwitches.

Open vSwitch, налаштований як традиційний комутатор рівня 2, успішно вдається переадресувати всі типи трафіку, за винятком пакетів багатоадресної передачі L2, таких як LLDP, STP, протокол агрегування посилань (LACP) та протокол Cisco discovery (CDP). Ці кадри використовують зарезервовані багатоадресні адреси призначення MAC у діапазонах 01: 80: c2: 00: 00: xx та 01: 00: 0c: cc: cc: xx, і, як правило, обробляються мостами Ethernet. З OVS, налаштованим як самостійний міст, прийнятно, щоб ці кадри оброблялися або скидались, а не заливалися в іншу частину мережі.

Стандартний комутатор рівня 2 - комутатор NETGEAR Fast Ethernet FS108 - також використовувався в якості базової лінії для порівняння з іншими результатами. Йому вдалося переадресувати всі типи трафіку, за винятком кількох кадрів багатоадресної передачі L2:

• Operations, Administration and Maintenance (OAM), протокол, який використовується для виявлення проблем з підключенням у мережах рівня 2;

• LACP;

Загалом, перевірені мережеві гіпервізори напрочуд добре передавали кілька типів трафіку. Єдині загальні проблеми, що спостерігаються, трапляються з трафіком IPv6, LLDP та STP. IPv6 не підтримується OpenFlow 1.0, який використовується як OVX, так і FlowVisor. LLDP і STP використовуються як протоколи управління мережами Ethernet. Хоча це обмеження очікується, оскільки ці протоколи призначені для обробки мережевими гіпервізорами, це потенційно серйозне обмеження для клієнтів, які хочуть запускати власні протоколи управління рівня 2 у своїх віртуальних мережах.

4.1.3 Відповідність стандартам адресації

4.1.3.1 Одноадресний перезапис заголовка IPv4

Щодо техніки перезапису заголовка, що використовується OVX, результати показують, що гіпервізор мережі замінює адреси IPv4 адресами в діапазонах X.0.0.0 / 8 для кожного хосту орендаря, де X - ідентифікатор орендаря, призначений під час конфігурації OVX. На рисунку 4.6 показаний знімок екрана захоплення пакета ICMP-запиту exo-сигналу від хосту A1 до A2 та відповіді від A2 до A1. Відстежуючи трафік мережі у порту 5 SW2, ми можемо спостерігати, що IP-адреса 10.0.0.1 переписується як 1.0.0.1, а IP-адреса 10.0.0.2 як 1.0.0.2.

Цікаво помітити, що якщо внутрішня та зовнішня IP-адреси однакові, OVX все ще встановлює правила потоку, які переписують IP-заголовок. Наприклад, коли хост C1 (3.0.0.1) надсилає запити exo-сигналу ICMP на хост C2 (3.0.0.2), OVX переписує їх адреси як 3.0.0.1 та 3.0.0.2 відповідно, як показано на рисунку 4.7. Це робиться тому, що хости C1 та C2 належать ідентифікатору клієнта 3, а їх внутрішні IP-адреси знаходяться в діапазоні 3.0.0.0/8.

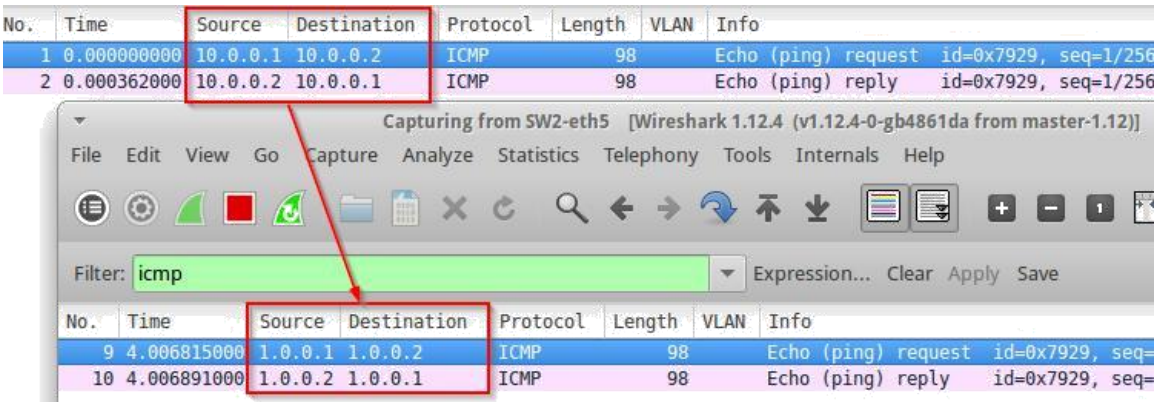
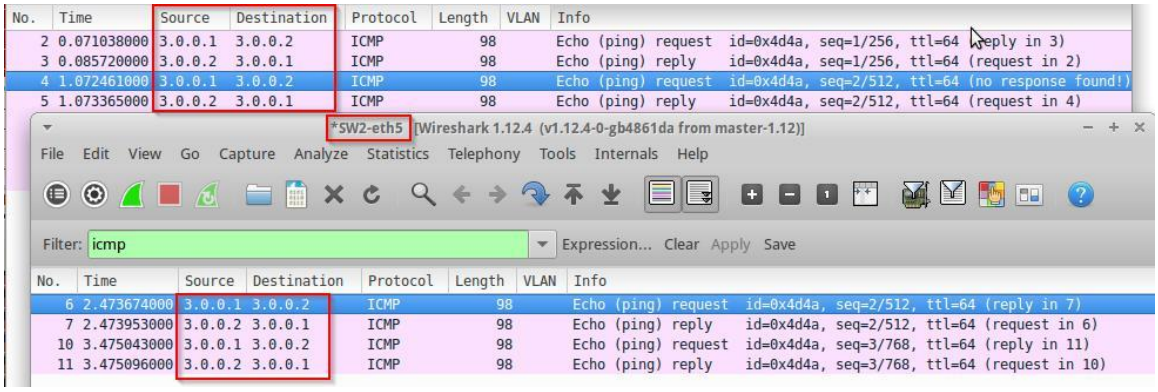


Рис. 4.6. Перезапис заголовка IP, виконаний OVX.



4.7. Перезапис заголовка IP, виконаний OVX з однаковими IP-адресами.

Повторювані адресні простори у внутрішній та зовнішній мережах, якими керує OVX, можуть значно ускладнити усунення несправностей у разі витоку пакетів із внутрішньої мережі, але насправді вони не створюють жодних проблем у можливостях пересилання трафіку віртуальної мережі. Однак внутрішня схема адресації OVX суперечить найкращим практикам адресації в інтернеті [53] і не може використовуватися поза повністю приватними мережами IPv4. Якщо будь-який трафік просочується з внутрішньої мережі в інтернет, він може потенційно переносити конфіденційні дані до невідомого місця призначення.

FlowVisor не має проблем із вирішенням питань дотримання стандартів, оскільки він не змінює трафік орендаря, а навпаки, гарантує, що кожен орендар володіє обмеженою частиною потокового простору.

4.1.3.2 Переписування ARP та багатоадресного заголовка

Щодо ARP, пакетів багатоадресної передачі IPv4 та багатоадресних кадрів MAC, OVX не представив жодних проблем із дотриманням вимог щодо пересилання цих кадрів через віртуальну мережу. Насправді OVX навіть не пересилав ці пакети через площину пересилання даних мережі. Всі кадри ARP, багатоадресні пакети IPv4 (адреса призначення 224.0.0.5) та багатоадресні адреси MAC (адреса призначення 01: 00: 0c: cc: cc: cc) завжди пересилаються через площину управління мережею. Це показано на рисунку 4.8., де кожен

PACKET_IN типу ARP або з адресатами багатоадресної передачі автоматично відправляється до всіх комутаторів через OVX за допомогою повідомлень PACKET_OUT. Це означає, що метод переписування заголовка, який використовує OVX, не застосовується до цих ситуацій.

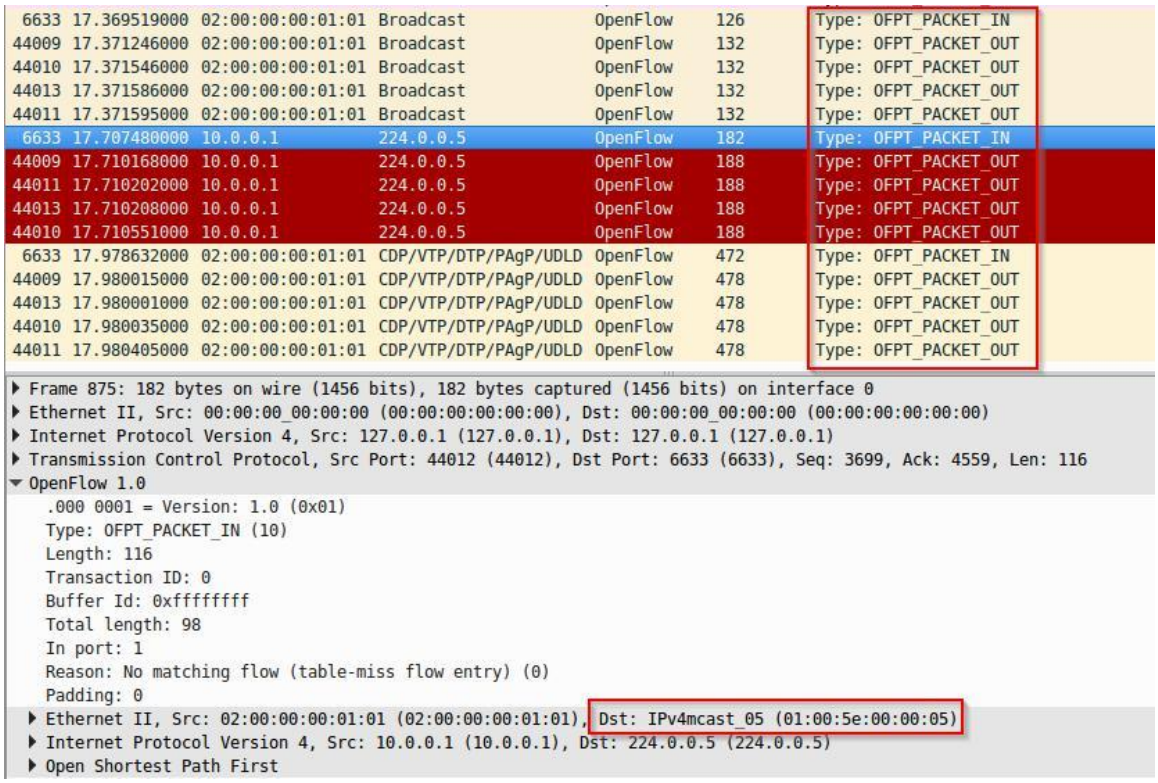


Рис.4.8. ARP, IP та MAC багатоадресна передача, залита OVX.

Незважаючи на те, що OVX та FlowVisor не мають проблем із дотриманням перевірених багатоадресних фреймів ARP та IP / MAC, результати показують, що обидва вони можуть спричинити проблеми в мережах багатоадресної передачі. Кожен окремий багатоадресний кадр, надісланий в мережу, обробляється мережевим гіпервізором окремо, ніколи не пересилаючи його контролеру орендаря Floodlight.

4.2 Результати перевірки продуктивності

Використовуючи фізичну топологію (розділ 3.1.2), тести продуктивності суттєво вплинули через неможливість комутатора встановити необхідні правила

FLOW MOD в апаратних таблицях. Через апаратні обмеження, комутатор OpenFlow міг використовувати лише таблиці потоків програмного забезпечення. Це створює величезний вплив на продуктивність результатів випробувань, особливо в тестах пропускної здатності.

Для всіх графіків вікон, показаних у наступних розділах, викиди (точки даних, представлені колами) були відкинуті при обчисленні статистичних даних. Викиди визначаються як точки даних, які виходять за межі, визначені рівняннями 4.1 та 4.2 для кожного набору даних. Q1 являє собою перший квартиль, Q3 представляє третій квартиль, а IQR представляє міжквартильний діапазон кожного набору даних.

Нижня межа = $Q_1 - 1.5 * IQR$ (4.1)

Верхня межа = $Q_3 + 1.5 * IQR$ (4.2)

4.2.1 Час налаштування потоку

Тести на затримку були можливі лише для FlowVisor та OVX. VeRTIGO викликав скидання перших пакетів потоку під час установки потоку. Тому тест пінгу не міг виміряти час налаштування потоку для VeRTIGO.

Цей експеримент проводився у фізичній топології, описаній у розділі 3.1.2, і перевіряв випадки встановлення часу потоку для:

- Floodlight;
- FlowVisor та Floodlight;
- Немає контролера, що використовує комутатор у традиційному режимі переадресації рівня 2;
- OVX та Floodlight.

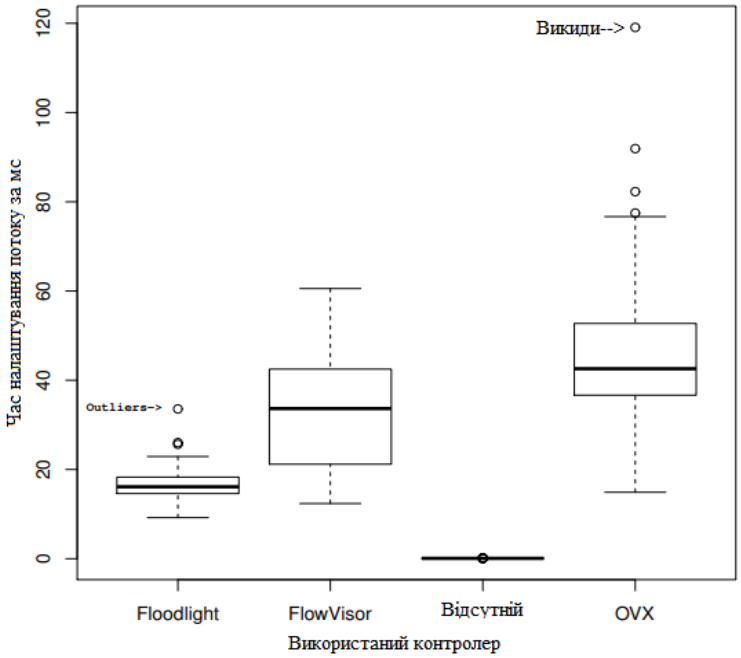


Рис. 4.9. Час затримки першого пінгу

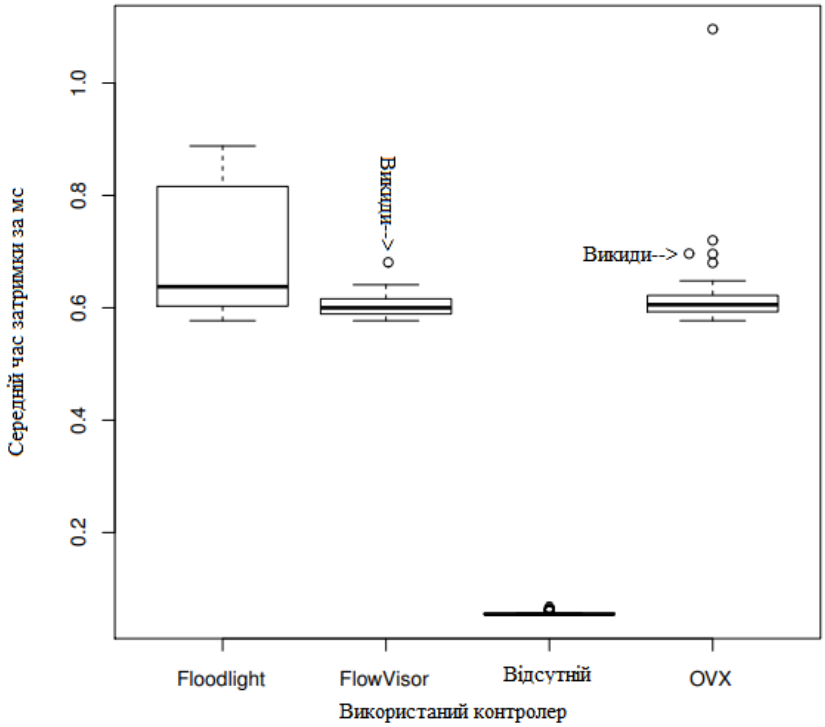


Рис. 4.10 Середній час затримки наступних пінгів

Як показано на рисунку 4.9., час налаштування потоку лише за допомогою Floodlight становив в середньому менше 20 мс, тоді як час налаштування потоку FlowVisor плавав близько 20-40 мс, а час налаштування потоку OVX - близько

40-50 мс. Дисперсія часу встановлення потоку була вищою у випадку OVX, і цей результат, ймовірно, пов'язаний з тим, що OVX покладається на складні правила потоку, щоб переписати трафік при вході або виході з мережі. Автономний комутатор рівня 2 давав в середньому 0,08 мс часу налаштування потоку. Це представлено контролером “Відсутній”. Ці результати відповідали очікуванням: час налаштування потоку набагато вищий при використанні мережевих гіпервізорів між мережею та контролерами орендаря, а програмно-визначені мережі демонструють значні накладні витрати на час налаштування потоку в порівнянні з традиційними мережами.

Рисунок 4.10 показує, що середній RTT пінгів, ініційованих після налаштування потоку, взагалі не залежить від контролера. Це очікується після встановлення потоків, оскільки контролер більше не впливає на затримку між хостами. Крім того, саме за цим сюжетом можна спостерігати різницю між програмним та апаратним режимами переадресації. Автономний режим переадресації комутатора контролером “Відсутній” використовує перевагу апаратної переадресації швидкості передачі, що призводить до приблизно в 10 разів швидших RTT і набагато менших дисперсіях.

4.2.2 Пропускна здатність

Тест пропускної здатності базується на iperf в режимі TCP. TCP являється найкращим протоколом, і пропускна здатність залежить від затримки мережі, розміру кадру, втрати пакетів, міжкадрового розриву та інших мережевих умов [10]. Очікувана пропускна здатність у локальній мережі з низькою затримкою повинна бути в межах 90-100% від діапазону автоматичного узгодження лінії зв'язку 100 Мбіт / с. Однак на результати тестів вплинула низька продуктивність комутатора HP-3500yl-24G, і виміряна пропускна здатність становила менше 1 Мбіт / с.

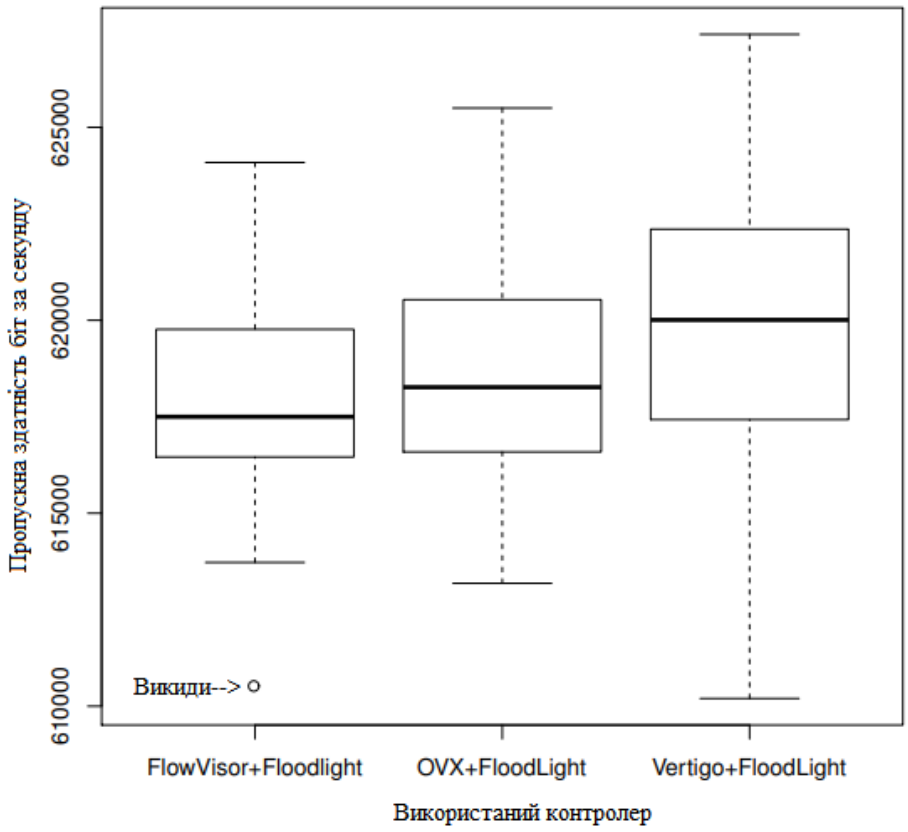


Рис.4.11 Пропускна здатність, виміряна різними контролерами в бітах за секунду

Результати експериментів, проведених у фізичній топології, зведені на рисунку 4.11. Пропускна здатність із FlowVisor в середньому становить 617,9 кбіт / с, OVX 618,6 кбіт / с та VeRTIGO 620,0 кбіт / с. Однак дисперсія у всіх результатах досить велика, і статистично не спостерігається значної різниці в отриманій смузі пропускання, виміряній між двома хостами при переході на різні гіпервізори мережі. Цей результат був очікуваним, оскільки вузьким місцем цього експерименту був процесор комутації. Висока завантаженість процесора під час експерименту є найімовірнішою причиною того, що в наборі даних «FlowVisor + Floodlight» є точка відхилення. Прості завдання під час експерименту, такі як відкриття сеансу telnet комутатору, можуть впливати на пропускну здатність, оскільки вони вимагають від ЦП тимчасово припинити обробку трафіку.

Додаткові дослідження та тести підтвердили, що обмеження продуктивності було встановлено обмеженим обладнанням HP-3500yl-24G [41]. Перегляд таблиць потоків комутатора також виявив, що навіть найпростіші правила потоку обробляються не апаратним забезпеченням переадресації, а його центральним процесором. Це означає, що комутатор HP-3500yl-24G не може навіть обробляти найосновніші правила потоку в апаратному забезпеченні.

Тести пропускної здатності повторювали за допомогою апаратного комутатора GENI, і результати були набагато задовільнішими. Рисунки 4.12. і 4.13 показують результати. Поки Floodlight, FlowVisor та VeRTIGO зуміли встановити потоки, які забезпечували пропускну здатність майже 100 Мбіт / с в тестах iperf TCP між хостами, пропускна здатність OpenVirteX в середньому становила приблизно 339 кбіт / с. Знову ж таки, TCP є найкращим протоколом і намагається використовувати якомога більше пропускної здатності. Пропускна здатність для цього експерименту також повинна залишатися між 90-100% пропускної здатності каналу (100 Мбіт / с).

Рисунок 4.12. також показує дві точки викиду даних. Перше відхилення знаходиться в стовпці "Тільки Floodlight ", точка даних, яка перевищує пропускну здатність каналу зв'язку в 100 Мбіт / с. Це значення не є реалістичним, і причина аномалії невідома. Другий випадок стався з «FlowVisor + Floodlight», і це показує, що одне вимірювання було значно нижчим, ніж решта. Це може означати декілька різних речей, таких як тимчасове перевантаження процесора в комп'ютері, на якому запущено iperf, або тимчасова втрата пакетів може спричинити механізм контролю перевантажень TCP. На щастя, відхилень було мало, і результати все ще вважаються дійсними.

Комутатори GENI не можуть увійти в систему, щоб підтвердити причину такого низького проходження через OVX, але швидше за все, відносно складні правила переписування заголовка, які вимагає OVX, не можуть оброблятись

апаратним забезпеченням цього комутатора. Точки викиди даних на рисунку 4.2.4 також вказують на те, що комутатор був перевантажений потоком TCP під час обробки пакетів з повільним процесором

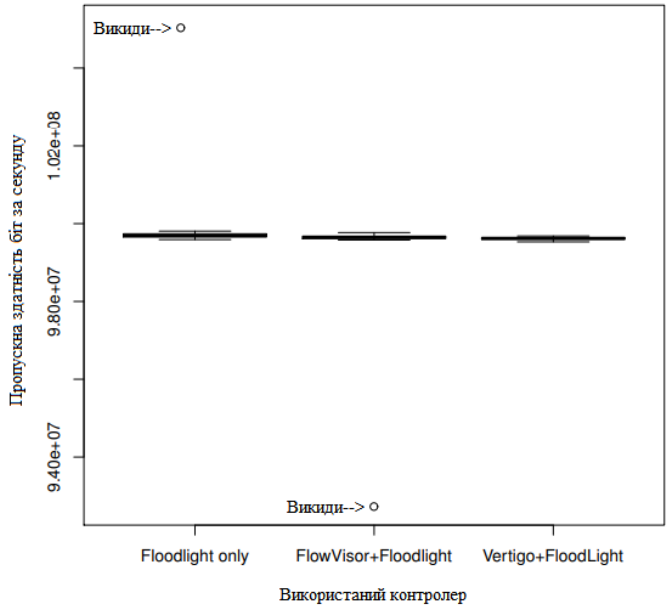


Рис.4.12. Пропускна здатність на тестовому стенді GENI, бітами в секунду

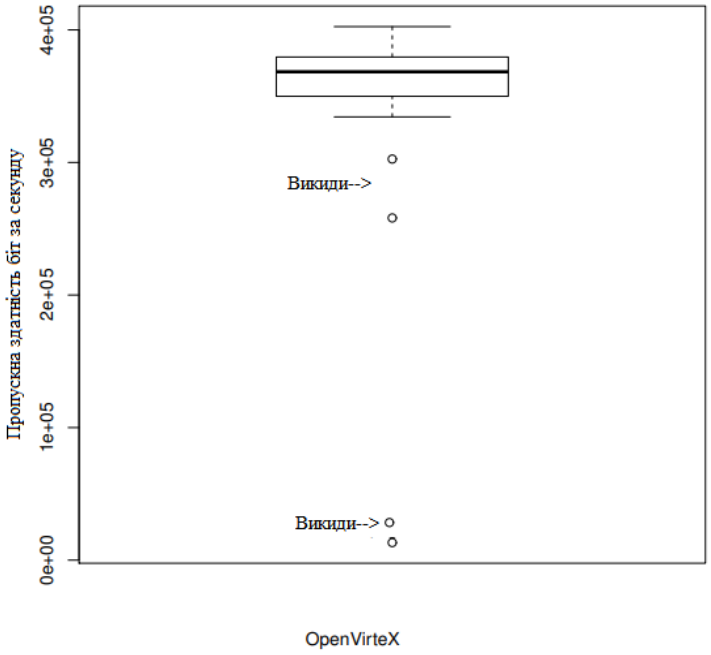


Рис.4.13. Пропускна здатність на тестовому стенді для OVX, бітами в секунду

4.3 Підсумки результатів та їх аналіз

Два із трьох додатків для віртуалізації мережі були успішно оцінені запропонованими тут експериментами. І FlowVisor, і OVX виправдали сподівання і протягом більшості експериментів працювали добре відповідно до своєї документації. Більшість експериментів з VeRTIGO не вдалося завершити через обмежену документацію та проблеми взаємодії з OVS та Floodlight. VeRTIGO мав кілька обмежень і був використаний лише для дуже простих тестів з апаратним комутатором. Спроби використовувати віртуальні посилання, щоб дозволити різні віртуальні топології, спричинили винятки у Floodlight при обміні повідомленнями з VeRTIGO.

Щодо ізоляції мережі та топології, FlowVisor реалізує простіший спосіб віртуалізації мережі порівняно з OVX. З одного боку, FlowVisor дозволяє розділяти зрізи на основі дуже гнучких обмежень, таких як цілий комутатор, фізичні порти, MAC-адреси або номери портів TCP / UDP. З іншого боку, OVX не допускає жодного руху в мережі без явного «підключення» хоста до однієї з його віртуальних мереж шляхом надання MAC-адреси та віртуального порту. Це означає, що адміністратор мережі - або програма, яка її контролює - повинна заздалегідь знати всі MAC-адреси, які будуть підключені до мережі.

FlowVisor не цікавить, використовує мережевий трафік IP чи ні, доки мережевий контролер здатний обробляти мережевий трафік, який піддається механізму нарізки FlowVisor. OVX, виходячи з поточного механізму роботи, передбачає, що контрольовані мережі працюватимуть з IPv4. Однак обидва засоби віртуалізації мережі обмежені можливостями OpenFlow, які вони підтримують. Наприклад, в даний час FlowVisor та OVX не підтримують IPv6, оскільки вони можуть обробляти лише OpenFlow 1.0, тоді як більшість реалізацій IPv6 пропонуються лише версіями 1.2 та 1.3 [40]. Це засвідчують експериментальні результати, представлені в розділі 4.1.3.

Як VeRTIGO, так і OVX стверджували, що мають підтримку автономного перенаправлення маршрутів у разі відмови каналу. VeRTIGO не можна було перевірити через проблеми сумісності. Функція автономного перенаправлення OVX спрацювала, але все ще не може гарантувати автономне відновлення, якщо хост клієнта не дозволить правилам потоку закінчитися, перш ніж намагатимется знову відправити трафік. Виміряти час відновлення мережі неможливо, оскільки функція перенаправлення залежить від використання мережі.

Що стосується стандартів адресації, прозорий режим роботи FlowVisor гарантує, що він готовий обробляти практично будь-який тип адресації, якщо він підтримується OpenFlow та додатком управління. Однак OVX майже виключно спирається на перезапис заголовка IPv4, щоб забезпечити функції мережевої ізоляції. Хоча він дуже добре працює в тестованих тут мережах, цей метод може бути проблемою в мережі IPv6, оскільки заголовки доведеться переписати з IPv6 на IPv4 і навпаки. Це може призвести до багатьох проблем сумісності. Іншим негативним аспектом призначення довільних мереж для переписування заголовків є те, що ізольований трафік може спричинити проблеми, якщо просочиться в Інтернет. Наприклад, OVX використовує внутрішні мережі, такі як 1.0.0.0/8 та 2.0.0.0/8, для ізоляції, але це дійсний загальнодоступний діапазон IP і ніколи не повинен використовуватися у приватних мережах.

Цікаво, що експеримент, який мав на меті перевірити відповідність стандартам, був корисним для знаходження серйозних обмежень у рівні віртуалізації мережі, запропонованих FlowVisor та OVX. Експерименти з ARP та багатоадресними кадрами показали, що перевірені мережеві гіпервізори обробляли ці кадри, завжди пересилаючи їх через площину управління, замість того, щоб встановлювати правила потоку для роботи з таким видом трафіку. Це може стати серйозним вузьким місцем у мережах з інтенсивним багатоадресним передаванням, оскільки кожен кадр багатоадресної передачі ARP або IP / MAC

повинен бути оброблений контролером перед відправкою в мережу. Крім того, рівень віртуалізації, накладений мережевими гіпервізорами, не передає ці кадри контролерам оренди. Натомість мережеві гіпервізори обробляють ці кадри локально, залишаючи контролерам орендарів невідомо, що коли-небудь існував багатоадресний фрейм, що рухався по мережі. Це означає, що контролер орендаря обмежений рівнем віртуалізації, навіть якщо він запрограмований на підтримку багатоадресної передачі.

Включення додаткових рівнів контролера віртуалізації мережі просто підкреслює вже існуючу проблему SDN. Час налаштування потоку вже визнано одним із недоліків OpenFlow (і SDN), оскільки централізований контролер дбає про пересилання рішень [62]. Традиційні мережі розподіленого рівня 2 мають дуже низький час налаштування потоку, часто кілька мілісекунд або навіть менше однієї мілісекунди. Це особливо важливий аспект для мереж, які потребують швидкого відновлення після відмови.

Тести пропускної здатності показали, що контролери проксі-сервера віртуалізації мережі можуть значно зменшити пропускну здатність, якщо встановлені потоки не підтримуються обладнанням пристрою. В експериментах, проведених у фізичній лабораторії, жодним з мережевих додатків не підтримувалося обладнанням OpenFlow, а всі контролери працювали погано через обмеження комутатора OpenFlow. Однак результати експериментів із використанням апаратного комутатора GENI показують, що лише віртуалізація OpenVirteX внесла значний вплив на продуктивність.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

Ця робота представила огляд літератури традиційних технологій віртуалізації мережі на основі SDN. Традиційні технології, такі як технології, засновані на використанні VLAN, MPLS та VxLAN, мають роки розвитку та стандартизації, що перетворюється на відносно стабільні рішення. Вони як і раніше корисні для вирішення більшості завдань віртуалізації, необхідних провайдерам Інтернет-послуг (ISP) та центрам обробки даних. Однак SDN забезпечує більшу гнучкість у вирішенні проблеми віртуалізації.

FlowVisor дозволяє адміністратору мережі нарізати мережу, використовуючи багато різних правил відповідності, дозволяючи різним додаткам спільно використовувати одну і ту ж мережу. Такої гнучкості важко досягти в традиційних мережах. Наприклад, як тільки хост підключений до порту за допомогою VLAN, існує не так багато способів надати різну обробку всьому трафіку, який позначений тим самим ідентифікатором VLAN. За допомогою FlowVisor різні способи обробки можуть бути легко надані різним портам додатків TCP або UDP. OVX надає послугу віртуалізації, подібну до тієї, яку можна досягти за допомогою VPN на основі MPLS. Всі хости, що належать одному орендодавцю, призначаються одній віртуальній мережі, і OVX обчислює найкращі маршрути всередині мережі для взаємозв'язку цих хостів, ніби вся мережа є великим комутатором. Це рішення схоже на послугу VPLS VPN, що надається мережами MPLS. З точки зору орендаря, мережа представляється єдиним великим комутатором. Перевага OVX полягає в тому, що конфігурація та управління мережею повністю централізовані, тоді як у традиційній мережі кожен комутатор повинен бути налаштований окремо.

На основі спостережень, зроблених протягом цієї роботи, FlowVisor найкраще підходить для роботи з різними програмами в рамках однієї мережі. Він може бути використаний як засіб віртуалізації мережі, щоб дозволити різним

клієнтам спільно використовувати ресурси переадресації рівня 2, забезпечуючи ізоляцію між зрізами. Однак його справжній потенціал полягає в тому, щоб надати кожному зрізу різну програму. Це означає, що набір комутаторів OpenFlow потенційно може бути використаний для забезпечення рішення переадресації на основі рівня 2 для одного клієнта, одночасно надаючи маршрутизацію IP іншому клієнту. З іншого боку, OVX в основному орієнтована на ізоляцію мережі. Він дуже добре підходить для використання в мережах центрів обробки даних, оскільки його механізм віртуалізації покладається на інформацію про порт і MAC-адреси, щоб визначити, до якої мережі належить хост. Незважаючи на те, що OVX сам по собі вимагає ручної конфігурації всіх хостів мережі, існують програмні розробки, які мають на меті інтегрувати OVX з OpenStack, надаючи забезпечення віртуальних машин разом із забезпеченням віртуальних мереж OVX через нейтронний плагін. По суті, це концепція інфраструктури як послуги (IaaS) [44].

Хоча підхід SDN є гнучким, він все одно має деякі недоліки. Як видно з результатів тесту продуктивності, час налаштування потоку є високим у мережах SDN і додатково збільшується, коли вводиться контролер проксі-сервера віртуалізації мережі. Доступні та перевірені тут рішення з відкритим кодом перебувають на ранніх стадіях розробки і можуть страждати від помилок та проблем сумісності, як це спостерігалося в експериментах з VeRTIGO або функцією автономної перенаправлення OVX. Ще одним негативним аспектом SDN є те, що методи віртуалізації взагалі не стандартизовані. Наприклад, якщо коли-небудь виникає необхідність з'єднати дві мережі від різних провайдерів за допомогою різних контролерів віртуалізації (наприклад, FlowVisor та OVX), слід звернути особливу обережність при обробці трафіку на краях мережі, щоб гарантувати, що мережі залишаться ізольованими належним чином.

Зіткнувшись з обробкою та прозорим пересиланням спеціальних типів трафіку, і FlowVisor, і OVX мали серйозні обмеження. Жоден з цих мережевих гіпервізорів не може переадресувати IPv6. Деякі протоколи управління рівня 2 блокуються рівнем мережевої віртуалізації, і багатоадресні програми повинні покладатися на інтенсивне перенаправлення площини управління процесором. Обмежуючи типи трафіку, який орендарі можуть використовувати в мережі, наявні в даний час методи віртуалізації на основі SDN стримують міграцію існуючих програм до нової парадигми SDN.

Творчий підхід OVX використовувати перезапис заголовків IP для забезпечення віртуалізації мережі напрочуд добре працює в експериментальних середовищах. Однак невибіркове використання довільних IP-адрес для здійснення віртуалізації мережі є сумнівним. Переписування заголовків IP робить процес усунення несправностей мережею більш складним, і ризики, пов'язані з використанням дійсних загальнодоступних IP-адрес для реалізації віртуалізації, можуть не коштувати витрат на потенційні витоки конфіденційних даних в Інтернет.

РОЗДІЛ 5 РОЗРОБКА СТАРТАП ПРОЕКТУ

5.1. Опис ідеї продукту

Таблиця 5.1.

Опис ідеї стартап проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Модернізація існуючих точок обміну трафіком шляхом впровадження технології SDN. Аудит роботи точок обміну трафіком, надання рекомендацій щодо покращення роботи мережевого SDN обладнання.	Великі та середні інтернет провайдери, телеком компанії.	Забезпечення стандартів якості обслуговування користувачів, збільшення контролю за мережею, спрощення організації, моніторингу та налаштування мережі

Таблиця 5.2.

Визначення сильних, слабких, та нейтральних характеристик ідеї проекту.

№ п/п	Техніко-економічні характеристики проекту	(потенційні) товари/концепції конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Embrane			
1	2	3	4	5	6	7
1	2	3	4	5	6	7
1.	Технічна характеристика	Створення мережевої інфраструктури точок обміну трафіком на основі технології SDN	Віртуалізація мережі, створення мережевих додатків	Доволі об’ємна робота для кожного окремого випадку немає готових рішень	Схожий підхід вирішення поставлених задач	Універсальність інструментів та методів при побудові мережі.

1	2	3	4	5	6	7
2.	Економічна характеристика	Виконання робіт проводиться в залежності від ТЗ та середня об’єктивна вартість створення мережевої інфраструктури точки обміну трафіком коливається близько 15000$	Проводиться повне дослідження очікуваного результату, проводяться роботи у повному обсязі Середня вартість від 20000 $	Великий об’єм роботи та тривала розробка проекту	немає	Великий вибір варіанту реалізації, економічна ефективність рішення

1	2	3	4	5	6	7
3.	Характеристика ефективності	Використання підходу з урахуванням можливостей та потреб замовника	Розробка здійснюється за шаблоном	Потребує більше часу на розробку	немає	Врахування потреб та можливостей замовника позитивно впливає на лояльність та розповсюдження реклами.
4.	Характеристика надійності	Використання технології SDN та віртуальних машин дозволяє мати повний контроль над системою та мережею	Розробка ведеться лише на основі шаблонів, та не дозволяє розкрити потенціал системи	Немає	Немає	Збільшує наійність системи, що в свою чергу збільшує лояльність замовника та зменшує витрати на гарантійне обслуговування системи

5.2. Технологічний аудит ідеї проекту

Таблиця 5.3.

Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Розробка мережі точки обміну трафіком базуючись на технології SDN	Інструменти технології SDN	Технологія наявна	Технологія доступна
2	Розробка системи моніторингу мережі	Інструменти протоколу OpenFlow	Технологія наявна	Технологія доступна
3	Використання віртуалізації для реалізації проекту	Технології VirtualBox, KVM, VMware	Налаштування необхідних параметрів на встановлених віртуальних машинах	Технологія доступна
Обрана технологія реалізації проекту: усі перераховані вище технології наявні та доступні. Реалізація потребує лише комбінування та налаштування вищевказаних технологій.				

5.3. Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 5.4.

Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	6
2	Загальний обсяг продаж, грн/ум.од.	500 тис ум.од.
3	Динаміка ринку	Зростає
4	Наявність обмежень для виходу	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Специфічні сертифікати підвищать можливий попит
6	Середня норма рентабельності в галузі, %	Більше 125%

За загальними оцінками ринок програмно-конфігурованих мереж є відкритим та зростаючим. Конкуренція присутня, але знаходиться на достатньо низькому рівні, тому можна вважати цей ринок придатним для інвестицій.

Таблиця 5.5.

Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних груп клієнтів	Вимоги споживачів до товару
1	Потреба у оновленні мережевого обладнання, підвищення якості, швидкості та контрольованості мережі, спрощення адміністрування мережі	Інтернет-провайдери, інтернет компанії виробники контенту, центри обробки даних	Відмінність полягає у розмірах мережі, та налаштування її на певний вид трафіку	Висока контрольованість та передбачуваність мережі Просте адміністрування мережі незалежно від виробника мережевого обладнання

Таблиця 5.6.

Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	2	3	4
1	Цінова конкуренція	Зміни в ціновій політиці компаній конкурентів	Пошук шляхів здешевлення системи, що розробляється

1	2	3	4
2	Потреба у нових технологіях у мережевій інфраструктурі	Зменшення потреби у нових технологія у сфері комп’ютерних мереж	Пошук шляхів популяризації програмно-конфігурованих мереж

Таблиця 5.7.

Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	2	3	4
1	Розвиток технологій у сфері програмно-конфігурованих мереж	Можливість покращення якості розроблених програмно-конфігурованих мереж, їх надійності, швидкодії	Оновлення методик та технологій які застосовуються для надання послуг, проведення навчання працівників новим технологіям.
2	Стрімких ріст ринку програмно-конфігурованих мереж	Можливість захватити більшу частину ринку, збільшення прибутку за рахунок обсягу	Нарощування потужності компанії для задоволення попиту, найм нових працівників.

Таблиця 5.8.

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Олігополістична конкуренція	Наявність на ринку невеликої кількості конкурентів	Можливість вийти на ринок за рахунок нових підходів
2. За рівнем конкурентної боротьби - міжнародний	Універсальність реалізації послуг закордоном	Можливість вийти на світовий ринок
3. За галузевою ознакою - внутрішньогалузева	Послуги є широко застосовуваними в одній сфері	Можливість брати замовлення через рекомендації
4. Конкуренція за видами товарів: - товарно-видова	Послуги, виконуються виходячи з потреб клієнтів, проте технологія відрізняється	Використовується комплексний підхід
5. За характером конкурентних переваг - цінова	Вартість послуг відрізняється	Збільшення цінності наданих послуг на одиницю грошей
6. За інтенсивністю - марочна	Визначаються підходи до надання сервісів	Збільшення кількості послуг, та покращення їх якості

Таблиця 5.9.

Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товаро-замінники
Складові аналізу	- Epsilon - Fortinet - Lightriver	Розмір вкладень у розвиток	Постачання апаратного забезпечення	Розміри запитуваних послуг, контроль якості наданих послуг	Витрати, що змінюються, собівартість послуг
Висновки	Конкуренція на ринку може вважатися малою, через малу кількість конкурентів	Присутня можливість виходу ринок при умові надання послуг достатньо високої якості	Мала впливовість постачальників на ринок, так як присутня мала залежність	Умови ринку повністю залежать від клієнтів	На ринку достатньо мала поширеність товарів-замінників

Присутня велика імовірність того, що даний проект буде продуктивний, так як, на ринку існує мала конкуренція, та наявна велика залежність якості продукту від технології, яка застосовувалась про проектуванні. Сильними сторонами проекту є використання нових технологій програмно-конфігурованих мереж, гнучке налаштування завдяки використанні цієї технології, малу присутність аналогічних послуг на ринку України. Також прогнозується стрімкий розвиток технології, що в майбутньому буде стимулювати розвиток та покращення послуг, що надаються.

Таблиця 5.10.

Обгрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обгрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Підхід до надання послуг	Розробка системи використовуючи унікальний підхід
2	Ціна	Конкурентна вартість послуг
3	Доступність ресурсів та технологій	Ресурси є у відкритому доступі, та визначаються підходом до надання послуг
4	Гнучкість у розробці	Максимальне підлаштовування під вимоги клієнта

Таблиця 5.11.

Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з 6point						
			-3	-2	-1	0	1	2	3
1	Підхід до надання послуг	18		+					
2	Ціна	19	+						
3	Доступність ресурсів та технологій	11				+			
4	Гнучкість у розробці	18		+					

Таблиця 5.12.

SWOT аналіз стартап проекту

Сильні сторони: гнучкість послуг під бажання клієнта, унікальний підхід до надання послуги	Слабкі сторони: малозабезпечена маркетингова сторона проекту, необхідна технічна підтримка виконаних проектів
Можливості: створення великої клієнтської бази, популяризація використання програмно-конфігурованих мереж, велика варіативність та легкість налаштування під бажання клієнта	Загрози: нестабільна ситуація на ринку, цінова конкуренція

Таблиця 5.13.

Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтований комплекс заходів) ринкової поведінки	Імовірність отримання результатів	Строки реалізації
1	Зменшення кількості використовуваних технологій	Висока імовірність залучення ресурсів	1.5 року
2	Відмова від нових та складних технологій та зосередження простоті швидкості впровадження	Мала імовірність залучення ресурсів	4 роки
3	Надання послуг на основі одної системи програмно-конфігурованих мереж	Висока імовірність залучення ресурсів	1 рік

На основі аналізу альтернативних варіантів впровадження проекту, було обрано підхід на основі використання одної системи програмно-конфігурованих мереж для покращення надійності та стабільності наданих послуг.

5.4. Розроблення ринкової стратегії проекту

Таблиця 5.14.

Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів прийняти продукт	Орієнтований попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу в сегмент
1	2	3	4	5	6
1	Інтернет-провайдери	Готовність висока через потребу у послугах, що пропонується	Високий попит	Висока конкуренція через відкритість технології та можливості клієнтів забезпечити себе ними	Складно, через новизну технології та малий обсяг вдало впроваджених проектів

1	2	3	4	5	6
2	Компанії – генератори контенту	Готовність висока через необхідність надавати контент з меншими видатками на мережу	Високий попит	Конкуренція невисока, так як дані компанії не спеціалізуються на створення та впровадженні мережевих технологій	Середня складність через відсутність великої маркетингової кампанії.
Які цільові групи обрано: обрано запропоновані цільові групи					

Беручи до уваги усі аспекти та особливості цільових груп було обрано стратегію диференційованого маркетингу.

Таблиця 5.15.

Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентноспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Флангова атака	Стратегія диференційованого маркетингу	Сильні сторони запропонованого рішення	Стратегія диференціації

Таблиця 5.16.

Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем на ринку?»	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Проект має унікальні частини	Поєднання пошуку нових та переманювання існуючих	Копіювання способу надання послуг	Стратегія флангової війни

Таблиця 5.17.

Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентно-спроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	2	3	4	5

1	2	3	4	5
1	Забезпечення високої пропускної здатності мережі, простий процес зміни у конфігурацію мережі «на ходу», Простий та зрозумілий моніторинг мережі, гнучке налаштування конфігурації під власні потреби	Стратегія диференціації	Новизна у технології та підході, універсальність рішень для клієнтів	Висока швидкість передачі, повний контроль над мережею, зрозумілий та простий моніторинг

5.5. Розроблення маркетингової програми стартап проекту

Таблиця 5.18.

Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	2	3	4
1	Швидкість маршрутизації трафіку	Збільшення пропускної здатності мережі за рахунок зменшення службового трафіку	Індивідуальний підхід при налаштування мережі
1	2	3	4
2	Надійність системи	Використання віртуалізації забезпечує	Висока надійність компонентів мережі

		надійність та резервування основних вузлів мережі	

Таблиця 5.19.

Визначення ключових переваг концепції потенційного товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Розробки мережевої інфраструктури за допомого технології SDN		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Швидкість мережі	М	Вр/Тх/Тл
	Надійність мережі	М	Тх/Тл/Е/Ор
	Впровадження нових технологій	М	Вр/Тх/Тл/Е/Ор
	Якість: забезпечення стабільності з’єднання, високої пропускної здатності		
	Пакет розробок інтегрується в систему		
	Марка: 6point		
III. Товар з підкріпленням	До продажу: для покращення продажів існує варіант тестового макету системи для демонстрацій		
	Після продажу: Проведення рекламної кампанії		
За рахунок чого потенційний товар буде захищено від копіювання: закриті розробки які захищені правом на інтелектуальну власність			

Таблиця 5.20.

Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи	Верхня та нижня межі встановлення ціни на товар/полсугу
1	25000 ум.од.	20000 ум.од	Високий	15000-20000 ум.од. в залежності від ступеню розміру мережі, вимог до швидкодії та пропускної здатності

Таблиця 5.21.

Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Необхідність оновлення мережі точок обміну трафіком	Неперервне покращення маркетингової стратегії, оновлення послуг відповідно до потреб ринку та нових розробок	Дворівневий канал збуту з використанням дистриб’юторів та використання сторонніх сервісів	Об’єднання зі сторонніми компаніями для формування широкої системи збуту

Таблиця 5.22.

Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Можливість моніторингу та тестування послуг, з детальними поясненнями функціонування різних рівнів мережі	Інтегрована маркетингова комунікація	Застосування технології SDN, віртуалізація, постійна підтримка користувачів, відсутність аналогів по співвідношенню ціна/якість	Популяризація програмно-конфігурованих мереж серед цільової аудиторії	Реклама, яка презентує новий підхід до проектування, налаштування та моніторингу мережі

Висновки

У даному розділі було запропоновано та розроблено стартап проект. Стратегія розгортання цього проекту базується на грунтовному вивченні ринку споживачів – інтернет провайдерів та підприємств, які генерують контент. Аналіз останніх досліджень ринку показав, що проект має високу шанси виходу на ринок, комерціалізації продукту, подальшого розвитку зі збільшенням капіталу та становлення як потужного рішення у сфері мережевих технологій. З огляду на малу конкуренцію та достатню конкурентноспроможність проекту, невеликий бар’єр входження є великі перспективи впровадження проекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Будылдина Н.В., Шувалов В.П. Сетевые технологии высокоскоростной передачи данных: [учебное пособие для вузов]/ под ред. ВП. Шувалова. – Москва: Горячая линия – Телеком, 2016г. – 343 с.: ил.
2. SDN&NFV [Электронный ресурс]/ Bellintegrator: Режим доступа: http://www.bellintegrator.ru/services-sdn-nfv.html
3. Барсков А. SDN: кому и зачем это надо?/ Журнал сетевых решений/LAN. 2012. № 12. – [Электронный ресурс]. – Режим доступа: https://www.osp.ru/ lan/2012/12/13033012
4. F. Pakzad, M. Portmann, W. L. Tan, and J. Indulska, “Efficient topology discovery in software defined networks,” in Signal Processing and Communication Systems (ICSPCS), 2014 8th International Conference on, Dec. 2014, pp. 1–8. doi: 10.1109/ICSPCS.2014.7021050
5. Mininet. (Feb. 2016). Mininet overview, [Online]. Available: http://mininet.org/ overview/ (visited on 01/20/2016).
6. M. Lewis, Comparing, Designing, and Deploying VPNs. Cisco Press, 2006. [Online]. Available: http://ptgmedia.pearsoncmg.com/images/1587051796/ samplechapter/1587051796content.pdf (visited on 03/12/2016).
7. “Framework for Layer 2 Virtual Private Networks (L2VPNs),” Tech. Rep., Sep. 2006. doi: 10.17487/rfc4664. [Online]. Available: https://tools.ietf.org/html/ rfc4664
8. Floodlight. (n.d.). Floodlight project, [Online]. Available: http://www.projectfl oodlight.org/
9. Felipe Stall Rechia «SDN Based Network Virtualization Techniques»
10. D. Drutskoy, E. Keller, and J. Rexford, “Scalable Network Virtualization in Software-Defined Networks,” Internet Computing, IEEE, vol. 17, no. 2, pp. 20– 27, Mar. 2013. doi: 10.1109/MIC.2012.144

11. R. Doriguzzi Corin, M. Gerola, R. Riggio, F. De Pellegrini, and E. Salvadori, “Vertigo: Network virtualization and beyond,” in Software Defined Networking (EWSDN), 2012 European Workshop on, Oct. 2012, pp. 24–29. doi: 10.1109/ EWSDN.2012.19.

12. OpenVirteX – Network Virtualization Platform[Online] https://github.com/os-libera/OpenVirteX

13. McKeown N. OpenFlow Enabling Innovation in Campus Networks / N. McKeown, T. Anderson // SIGCOMM. – 2008. – Vol. 38. – P. 69- 74. 100

14. Стеклов В. К. Проектування телекомунікаційних мереж / В. К. Стеклов, Л. Н. Беркман. ; під ред. В. К. Стеклова – Київ : Техніка, 2002. – 792 с. 21.

15. Стеклов В. К. Сучасні системи управління в телекомунікація / В. К. Стеклов, Б. Я. Костік, Л. Н. Беркман ; під ред. В. К. Стеклова – Київ : Техніка, 2005. – 395 с.

16. Stallings W. Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud / Stallings W. — CA.: published by Pearson, 2015. – 116 с.

17. Doherty J. SDN and NFV Simplified / Doherty J. — CA.: published by Addison-Wesley Professional, 2016. – 173 с. 10.

18. Tiwari V. SDN and OpenFlow for Beginners with Hands on Labs / Tiwari V. — CA.: published by Amazon Digital Services LLC, 2013. – 15 с.

19. OpenFlow Tutorial//OpenFlow.2013.URL. //http://archive.openflow.org/wk/index.php/ OpenFlow_Tutorial (Дата обращения: 07.11.2013).

20. ONF Specification//Open network foundation.2013.URL: https://www.opennetworking.org/sdn-resources/onf-specifications (дата обращения: 07.11.2013).

21. O N. M. M. K. Chowdhury and R. Boutaba, “A survey of network

virtualization," Computer Networks, vol. 54, no. 5, pp. 862–876, 2010. doi: http://dx.doi.org/ 10.1016/j.comnet.2009.10.017. [Online]. Available: http://www.sciencedirect. com/science/article/pii/S1389128609003387.

22. R. D. Corin and M. Gerola. (Nov. 2013). Vertigo @ github.com, [Online]. Available: https://github.com/fp7-ofelia/VeRTIGO (visited on 11/25/2015).

23. M. De Leenheer. (Dec. 2015). Openvirtex discussion forum - can ovx discover the hosts information? [Online]. Available: https://goo.gl/hL8yTe (visited on 03/11/2016).

24. S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," Tech. Rep., Dec. 1998. doi: 10.17487/rfc2460. [Online]. Available: https://tools. ietf.org/html/rfc2460.

25. R. Doriguzzi Corin, M. Gerola, R. Riggio, F. De Pellegrini, and E. Salvadori, "Vertigo: Network virtualization and beyond," in Software Defined Networking (EWSDN), 2012 European Workshop on, Oct. 2012, pp. 24–29. doi: 10.1109/ EWSDN.2012.19.

26. D. Drutskoy, E. Keller, and J. Rexford, "Scalable Network Virtualization in Software-Defined Networks," Internet Computing, IEEE, vol. 17, no. 2, pp. 20– 27, Mar. 2013. doi: 10.1109/MIC.2012.144.

27. D. Erickson, "The Beacon OpenFlow Controller," in Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, ser. HotSDN '13, Hong Kong, China: ACM, 2013, pp. 13–18. doi: 10.1145/2491185. 2491189. [Online]. Available: http://doi.acm.org/10.1145/2491185.2491189.

28. R. R. Hain. (Mar. 2015). Intro to openflow tutorial (hardware switch), [Online]. Available: http://groups.geni.net/geni/wiki/GENIExperimenter/Tutorials/ OpenFlowHW/DesignSetup

29. A. A. Jaha, F. B. Shatwan, and M. Ashibani, "Proper Virtual Private Network (VPN) Solution," in 2008 The Second International Conference on Next

Gener-ation Mobile Applications, Services, and Technologies, Institute of Electrical & Electronics Engineers (IEEE), 2008. doi: 10.1109/ngmast.2008.18.

30. Wireshark. (n.d.). Ethernet vendor codes, and well-known MAC addresses, [Online]. Available: https://code.wireshark.org/review/gitweb?p=wireshark.git; a=blob_plain;f=manuf .

31. A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sherwood, "On Controller Performance in Software-Defined Networks," in Presented as part of the 2nd USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services, San Jose, CA: USENIX, 2012. [Online]. Available: https://www.usenix.org/conference/hot-ice12-0/controller-performance-software-defined-networks

32. Tcpdump. (n.d.). Tcpdump Command-line Packet Analyzer, [Online]. Available: http://www.tcpdump.org/

33. R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Flowvisor: a network virtualization layer," Deutsche Telekom Inc. R&D Lab, Stanford, Nicira Networks, Tech. Rep., 2009

34. R. Sherwood. (2014). Cbench: A Benchmarking Tool for Controllers, [Online]. Available: https://github.com/andi- bigswitch/oflops/tree/master/cbench

35. M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright, "Virtual eXtensible local area network (VXLAN): A framework for overlaying virtualized layer 2 networks over layer 3 networks," Tech. Rep., Aug. 2014. doi: 10 . 17487 / rfc7348. [Online]. Available: https : //tools.ietf.org/html/rfc7348.

36. Mininet GitHub, [Online]. Available: git://github.com/mininet/ mininet